# High-Performance GPU-Accelerated Finite Element Analysis

**Annex A - Solver Implementations Detailed Report**

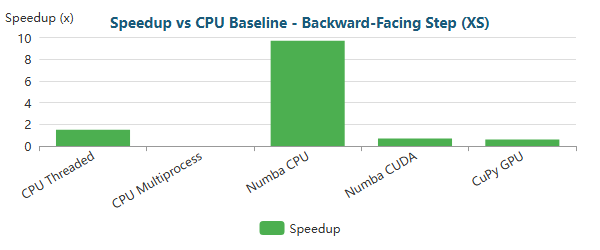
## 1. Solver Implementations Detailed Report

### 1.1 RTX 5090 Performance

Key results from performance benchmarks comparing FEM solver implementations.

**Backward-Facing Step (XS)** (287 nodes)

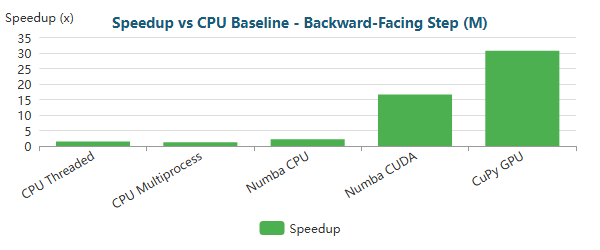
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 42ms ± 3ms | 1.0x | 3 |
| CPU Threaded | 27ms ± 1ms | 1.5x | 3 |
| CPU Multiprocess | 3.48s ± 0.02s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 9.7x | 3 |
| Numba CUDA | 57ms ± 1ms | 0.7x | 3 |
| CuPy GPU | 67ms ± 2ms | 0.6x | 3 |



Bar

**Backward-Facing Step (M)** (195,362 nodes)

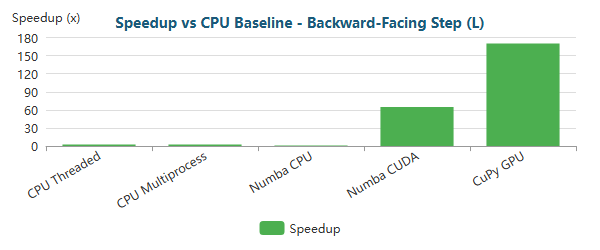
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 47.35s ± 0.06s | 1.0x | 3 |
| CPU Threaded | 30.68s ± 0.13s | 1.5x | 3 |
| CPU Multiprocess | 40.14s ± 0.11s | 1.2x | 3 |
| Numba CPU | 21.96s ± 0.95s | 2.2x | 3 |
| Numba CUDA | 2.86s ± 0.06s | 16.6x | 3 |
| CuPy GPU | 1.54s ± 0.00s | 30.7x | 3 |



Bar

**Backward-Facing Step (L)** (766,088 nodes)

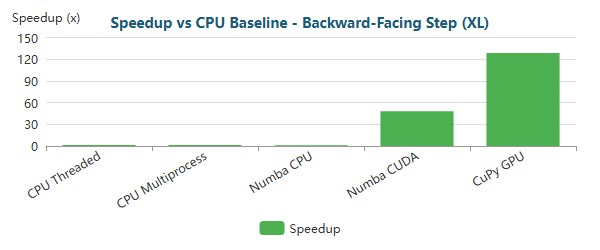
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 10m 12.4s ± 4.7s | 1.0x | 3 |
| CPU Threaded | 3m 38.3s ± 1.0s | 2.8x | 3 |
| CPU Multiprocess | 3m 48.2s ± 5.3s | 2.7x | 3 |
| Numba CPU | 8m 55.7s ± 4.0s | 1.1x | 3 |
| Numba CUDA | 9.48s ± 0.26s | 64.6x | 3 |
| CuPy GPU | 3.60s ± 0.01s | 170.1x | 3 |



Bar

**Backward-Facing Step (XL)** (1,283,215 nodes)

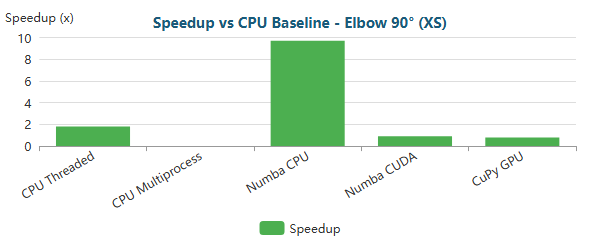
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 12m 8.6s ± 1.4s | 1.0x | 3 |
| CPU Threaded | 7m 50.4s ± 0.4s | 1.5x | 3 |
| CPU Multiprocess | 8m 37.0s ± 4.4s | 1.4x | 3 |
| Numba CPU | 9m 57.2s ± 1.3s | 1.2x | 3 |
| Numba CUDA | 15.13s ± 0.05s | 48.2x | 3 |
| CuPy GPU | 5.67s ± 0.00s | 128.5x | 3 |



Bar

**Elbow 90° (XS)** (411 nodes)

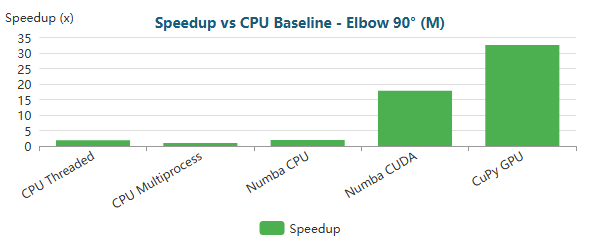
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 53ms ± 2ms | 1.0x | 3 |
| CPU Threaded | 29ms ± 4ms | 1.8x | 3 |
| CPU Multiprocess | 4.61s ± 0.02s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 9.7x | 3 |
| Numba CUDA | 57ms ± 5ms | 0.9x | 3 |
| CuPy GPU | 70ms ± 4ms | 0.8x | 3 |



Bar

**Elbow 90° (M)** (161,984 nodes)

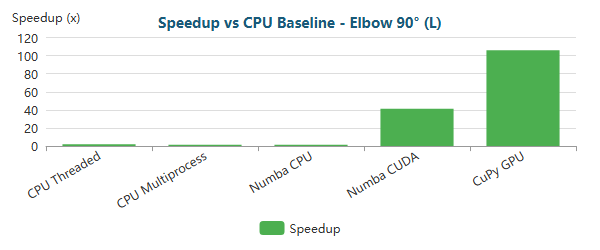
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 41.78s ± 0.09s | 1.0x | 3 |
| CPU Threaded | 22.37s ± 0.03s | 1.9x | 3 |
| CPU Multiprocess | 40.06s ± 0.16s | 1.0x | 3 |
| Numba CPU | 20.47s ± 0.80s | 2.0x | 3 |
| Numba CUDA | 2.35s ± 0.09s | 17.8x | 3 |
| CuPy GPU | 1.28s ± 0.00s | 32.6x | 3 |



Bar

**Elbow 90° (L)** (623,153 nodes)

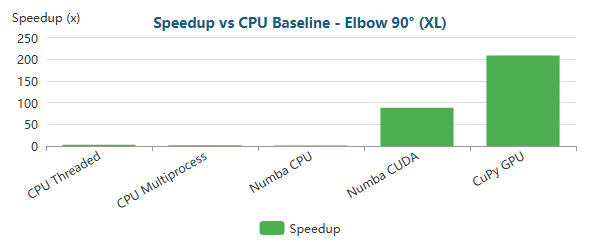
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 4m 52.6s ± 1.9s | 1.0x | 3 |
| CPU Threaded | 2m 22.8s ± 0.6s | 2.0x | 3 |
| CPU Multiprocess | 3m 28.5s ± 0.9s | 1.4x | 3 |
| Numba CPU | 3m 35.8s ± 2.2s | 1.4x | 3 |
| Numba CUDA | 7.08s ± 0.10s | 41.3x | 3 |
| CuPy GPU | 2.77s ± 0.01s | 105.8x | 3 |



Bar

**Elbow 90° (XL)** (1,044,857 nodes)

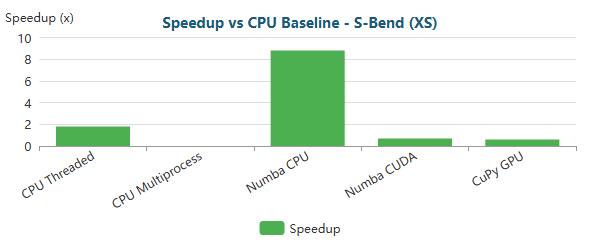
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 14m 25.9s ± 1.5s | 1.0x | 3 |
| CPU Threaded | 5m 0.3s ± 1.6s | 2.9x | 3 |
| CPU Multiprocess | 7m 31.4s ± 9.9s | 1.9x | 3 |
| Numba CPU | 13m 20.1s ± 6.2s | 1.1x | 3 |
| Numba CUDA | 9.82s ± 0.30s | 88.2x | 3 |
| CuPy GPU | 4.15s ± 0.01s | 208.6x | 3 |



Bar

**S-Bend (XS)** (387 nodes)

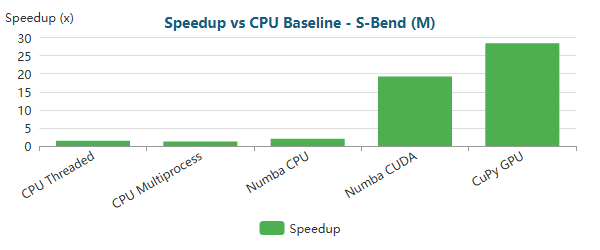
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 42ms ± 3ms | 1.0x | 3 |
| CPU Threaded | 24ms ± 3ms | 1.8x | 3 |
| CPU Multiprocess | 3.48s ± 0.03s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 8.8x | 3 |
| Numba CUDA | 62ms ± 6ms | 0.7x | 3 |
| CuPy GPU | 74ms ± 2ms | 0.6x | 3 |



Bar

**S-Bend (M)** (196,078 nodes)

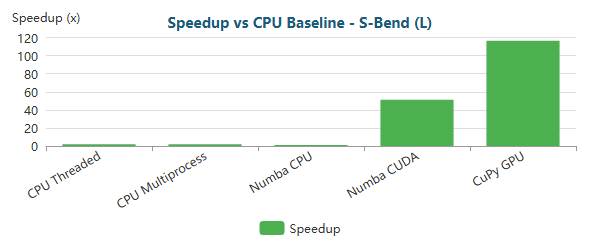
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 51.71s ± 0.32s | 1.0x | 3 |
| CPU Threaded | 33.86s ± 0.30s | 1.5x | 3 |
| CPU Multiprocess | 41.14s ± 0.13s | 1.3x | 3 |
| Numba CPU | 26.05s ± 1.28s | 2.0x | 3 |
| Numba CUDA | 2.70s ± 0.02s | 19.2x | 3 |
| CuPy GPU | 1.82s ± 0.01s | 28.4x | 3 |



Bar

**S-Bend (L)** (765,441 nodes)

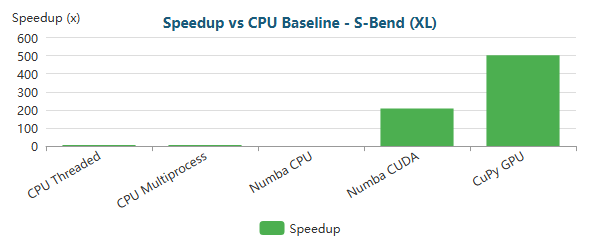
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 8m 10.3s ± 1.4s | 1.0x | 3 |
| CPU Threaded | 4m 4.5s ± 0.3s | 2.0x | 3 |
| CPU Multiprocess | 3m 58.4s ± 6.0s | 2.1x | 3 |
| Numba CPU | 6m 39.9s ± 3.8s | 1.2x | 3 |
| Numba CUDA | 9.57s ± 0.23s | 51.2x | 3 |
| CuPy GPU | 4.21s ± 0.02s | 116.4x | 3 |



Bar

**S-Bend (XL)** (1,286,039 nodes)

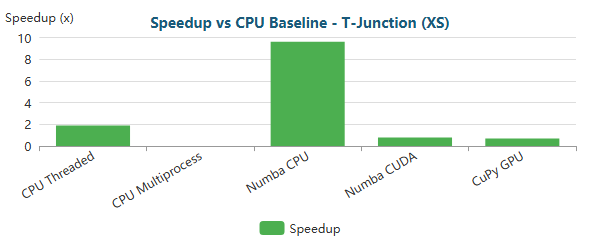
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 56m 1.2s ± 5.3s | 1.0x | 3 |
| CPU Threaded | 8m 51.3s ± 0.4s | 6.3x | 3 |
| CPU Multiprocess | 9m 2.5s ± 0.4s | 6.2x | 3 |
| Numba CPU | 58m 46.3s ± 34.9s | 1.0x | 3 |
| Numba CUDA | 16.15s ± 0.11s | 208.1x | 3 |
| CuPy GPU | 6.69s ± 0.01s | 502.2x | 3 |



Bar

**T-Junction (XS)** (393 nodes)

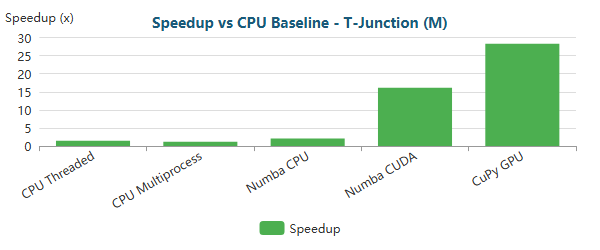
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 54ms ± 2ms | 1.0x | 3 |
| CPU Threaded | 29ms ± 5ms | 1.9x | 3 |
| CPU Multiprocess | 4.32s ± 0.02s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 9.6x | 3 |
| Numba CUDA | 64ms ± 2ms | 0.8x | 3 |
| CuPy GPU | 76ms ± 1ms | 0.7x | 3 |



Bar

**T-Junction (M)** (196,420 nodes)

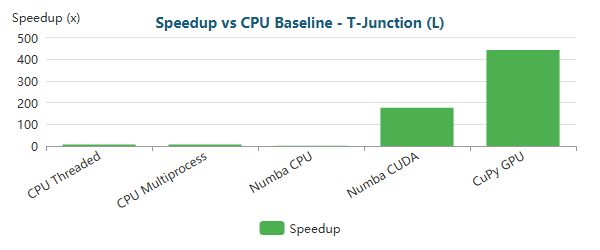
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 47.93s ± 0.03s | 1.0x | 3 |
| CPU Threaded | 32.61s ± 0.14s | 1.5x | 3 |
| CPU Multiprocess | 39.74s ± 0.29s | 1.2x | 3 |
| Numba CPU | 22.85s ± 0.94s | 2.1x | 3 |
| Numba CUDA | 2.98s ± 0.05s | 16.1x | 3 |
| CuPy GPU | 1.69s ± 0.00s | 28.3x | 3 |



Bar

**T-Junction (L)** (768,898 nodes)

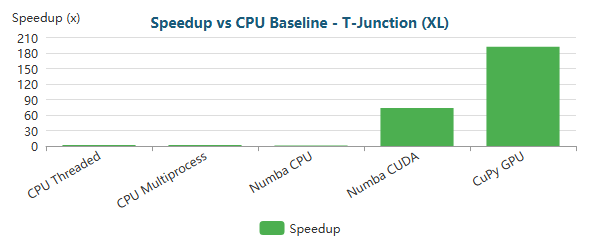
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 28m 33.7s ± 12.2s | 1.0x | 3 |
| CPU Threaded | 3m 50.4s ± 1.9s | 7.4x | 3 |
| CPU Multiprocess | 3m 45.8s ± 0.3s | 7.6x | 3 |
| Numba CPU | 23m 28.7s ± 4.3s | 1.2x | 3 |
| Numba CUDA | 9.71s ± 0.02s | 176.6x | 3 |
| CuPy GPU | 3.87s ± 0.01s | 442.8x | 3 |



Bar

**T-Junction (XL)** (1,291,289 nodes)

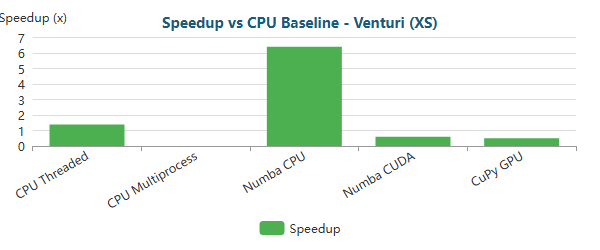
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 19m 6.6s ± 2.6s | 1.0x | 3 |
| CPU Threaded | 8m 17.4s ± 1.1s | 2.3x | 3 |
| CPU Multiprocess | 8m 28.3s ± 1.1s | 2.3x | 3 |
| Numba CPU | 16m 24.2s ± 1.9s | 1.2x | 3 |
| Numba CUDA | 15.55s ± 0.27s | 73.7x | 3 |
| CuPy GPU | 5.97s ± 0.01s | 192.0x | 3 |



Bar

**Venturi (XS)** (341 nodes)

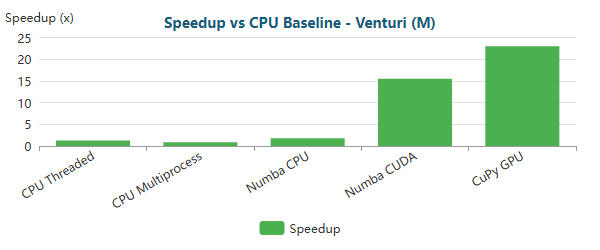
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 33ms ± 4ms | 1.0x | 3 |
| CPU Threaded | 23ms ± 1ms | 1.4x | 3 |
| CPU Multiprocess | 3.52s ± 0.02s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 6.4x | 3 |
| Numba CUDA | 54ms ± 4ms | 0.6x | 3 |
| CuPy GPU | 67ms ± 2ms | 0.5x | 3 |



Bar

**Venturi (M)** (194,325 nodes)

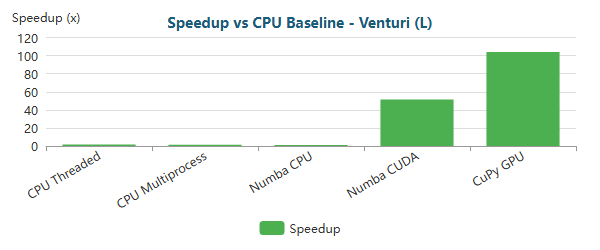
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 39.42s ± 0.04s | 1.0x | 3 |
| CPU Threaded | 30.84s ± 0.10s | 1.3x | 3 |
| CPU Multiprocess | 43.46s ± 0.88s | 0.9x | 3 |
| Numba CPU | 21.67s ± 0.21s | 1.8x | 3 |
| Numba CUDA | 2.55s ± 0.05s | 15.5x | 3 |
| CuPy GPU | 1.71s ± 0.03s | 23.0x | 3 |



Bar

**Venturi (L)** (763,707 nodes)

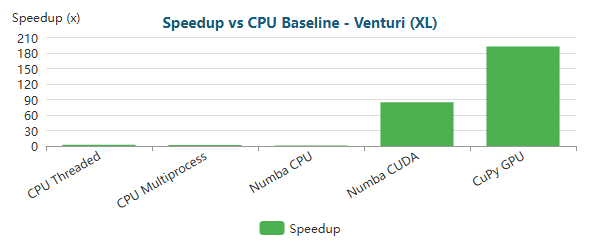
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 34.6s ± 5.3s | 1.0x | 3 |
| CPU Threaded | 3m 30.5s ± 0.8s | 1.9x | 3 |
| CPU Multiprocess | 4m 13.6s ± 6.3s | 1.6x | 3 |
| Numba CPU | 5m 49.2s ± 1.9s | 1.1x | 3 |
| Numba CUDA | 7.65s ± 0.04s | 51.6x | 3 |
| CuPy GPU | 3.80s ± 0.04s | 103.9x | 3 |



Bar

**Venturi (XL)** (1,284,412 nodes)

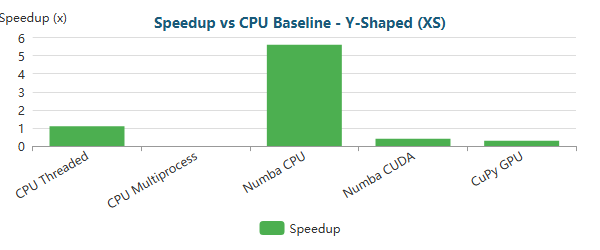
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 18m 35.3s ± 8.8s | 1.0x | 3 |
| CPU Threaded | 7m 21.7s ± 0.7s | 2.5x | 3 |
| CPU Multiprocess | 8m 47.5s ± 8.1s | 2.1x | 3 |
| Numba CPU | 15m 41.3s ± 1.5s | 1.2x | 3 |
| Numba CUDA | 13.14s ± 0.51s | 84.9x | 3 |
| CuPy GPU | 5.78s ± 0.03s | 192.8x | 3 |



Bar

**Y-Shaped (XS)** (201 nodes)

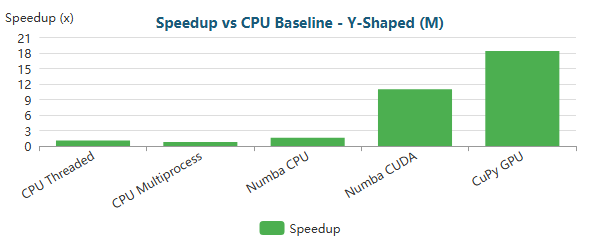
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 22ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 20ms ± 2ms | 1.1x | 3 |
| CPU Multiprocess | 2.45s ± 0.19s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 5.6x | 3 |
| Numba CUDA | 52ms ± 2ms | 0.4x | 3 |
| CuPy GPU | 65ms ± 5ms | 0.3x | 3 |



Bar

**Y-Shaped (M)** (195,853 nodes)

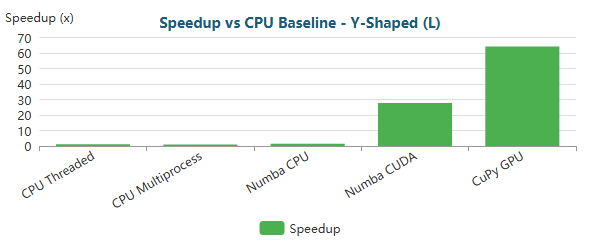
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 29.39s ± 0.65s | 1.0x | 3 |
| CPU Threaded | 25.92s ± 0.01s | 1.1x | 3 |
| CPU Multiprocess | 36.81s ± 0.82s | 0.8x | 3 |
| Numba CPU | 17.85s ± 0.18s | 1.6x | 3 |
| Numba CUDA | 2.68s ± 0.10s | 11.0x | 3 |
| CuPy GPU | 1.60s ± 0.04s | 18.4x | 3 |



Bar

**Y-Shaped (L)** (772,069 nodes)

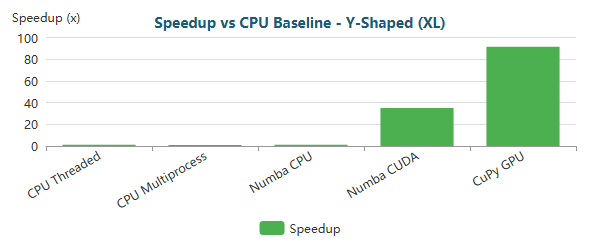
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 3m 44.8s ± 1.7s | 1.0x | 3 |
| CPU Threaded | 3m 6.8s ± 0.9s | 1.2x | 3 |
| CPU Multiprocess | 3m 20.3s ± 3.0s | 1.1x | 3 |
| Numba CPU | 2m 34.7s ± 2.7s | 1.5x | 3 |
| Numba CUDA | 8.09s ± 0.21s | 27.8x | 3 |
| CuPy GPU | 3.50s ± 0.04s | 64.2x | 3 |



Bar

**Y-Shaped (XL)** (1,357,953 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 8m 22.5s ± 18.4s | 1.0x | 3 |
| CPU Threaded | 7m 14.5s ± 0.9s | 1.2x | 3 |
| CPU Multiprocess | 7m 36.8s ± 3.6s | 1.1x | 3 |
| Numba CPU | 6m 18.7s ± 2.6s | 1.3x | 3 |
| Numba CUDA | 14.33s ± 0.22s | 35.1x | 3 |
| CuPy GPU | 5.49s ± 0.05s | 91.5x | 3 |



Bar

### 1.1.1 Critical Analysis RTX 5090

#### 1.1.1.1 Small-Scale Problems (XS meshes)

Across all geometries with **XS meshes (≈200–400 nodes)**, the results show a **clear and consistent pattern**:

* **CPU-based implementations dominate** absolute performance.
* **Numba JIT CPU** delivers the best results, with speedups typically ranging from **~4× to ~10×** relative to the baseline.
* **GPU implementations (Numba CUDA and CuPy GPU)** are systematically *slower* than the CPU baseline, with speedups in the range of **~0.3× to ~0.9×**, corresponding to real slowdowns.

This behavior is expected from a performance-modeling perspective. At this scale, FEM execution is dominated by fixed overheads, including kernel launch latency, host–device memory transfers, and GPU synchronization costs. These overheads cannot be amortized when the number of elements is small, even on a high-end accelerator such as the RTX 5090. As a result, GPU parallelism remains underutilized, confirming that **GPU acceleration is not suitable for small FEM problems**, regardless of hardware capability.

In addition, **CPU multiprocessing performs extremely poorly** in this regime, often taking several seconds, as process creation and inter-process communication dominate execution time. This reinforces that parallel execution models must be carefully matched to problem size.

#### 1.1.1.2 Medium-Scale Problems (M meshes)

For **medium-scale meshes (≈160k–200k nodes)**, the performance profile changes sharply and consistently across all geometries:

* **GPU acceleration becomes dominant**, marking a clear CPU–GPU crossover point.
* **Numba CUDA** achieves speedups of approximately **~11× to ~20×**.
* **CuPy GPU (RawKernel)** consistently outperforms Numba CUDA, reaching speedups between **~18× and ~33×**.
* **CPU-based approaches saturate**, rarely exceeding **~2× speedup**, even with threading, multiprocessing, or JIT compilation.

At this scale, arithmetic intensity and parallel workload are sufficient to fully exploit the RTX 5090’s massive parallelism and memory bandwidth. Assembly and post-processing costs become negligible, and total runtime is increasingly dominated by the sparse linear solver. More complex geometries (e.g., T-Junction and Venturi) benefit disproportionately from GPU execution, indicating improved efficiency as solver workload and sparsity complexity increase.

#### 1.1.1.3 Large-Scale and Extreme-Scale Problems (L and XL meshes)

For **large (L) and extra-large (XL) meshes (≈700k–1.35M nodes)**, GPU acceleration becomes **essential rather than optional**:

* **CPU baseline runtimes grow to several minutes**, making CPU-only execution impractical.
* **Threading and multiprocessing offer limited relief**, typically capped at **~2×–7× speedup**, and may degrade at XL scale due to memory pressure and synchronization overhead.
* **Numba JIT CPU loses effectiveness**, frequently approaching baseline performance as memory bandwidth becomes the dominant limitation.
* **Numba CUDA achieves speedups of ~40×–90×**, depending on geometry and scale.
* **CuPy GPU defines the performance envelope**, reaching **~100×–500× speedup** for the largest meshes.

At these scales, GPU execution effectively eliminates assembly and post-processing as bottlenecks. However, the sparse iterative solver becomes dominant, and performance is constrained primarily by memory bandwidth and sparse access patterns rather than compute throughput. The flattening of speedup curves at XL scale reflects this **solver-dominated, memory-bound regime**.

#### 1.1.1.4 Comparative Assessment Across Scales

From a practical standpoint, the results support the following execution-model selection:

* **XS meshes:** Numba JIT CPU — minimal overhead and compiled execution.
* **M meshes:** CuPy GPU (RawKernel) — optimal CPU–GPU crossover efficiency.
* **L and XL meshes:** CuPy GPU (RawKernel) — maximum scalability and throughput.
* **CPU-only environments:** Numba JIT CPU — best balance of speed and portability.
* **GPU prototyping:** Numba CUDA — faster development with acceptable performance.
* **Production GPU workloads:** CuPy RawKernel — highest and most consistent speedups.

#### 1.1.1.5 Comparative Assessment of Execution Models

From a practical standpoint, the benchmark results support the following conclusions:

| Regime | Best Execution Model | Rationale |
| --- | --- | --- |
| XS meshes | Numba JIT CPU | Minimal overhead, compiled execution |
| M meshes | CuPy GPU (RawKernel) | Maximum throughput, full GPU residency |
| CPU-only environments | Numba JIT CPU | Best balance of speed and memory efficiency |
| GPU prototyping | Numba CUDA | Easier development, acceptable performance |
| Production GPU | CuPy RawKernel | Highest and most consistent speedups |

The RTX 5090 demonstrates **excellent scalability** once the problem size justifies GPU usage. However, the results also highlight that **hardware capability alone is insufficient**: algorithmic structure, execution model, and problem scale are decisive factors in achieving high performance.

Overall, the benchmarks confirm that the RTX 5090 is exceptionally well suited for **large-scale FEM simulations**, delivering order-of-magnitude speedups over CPU execution when properly utilized. At the same time, the data reinforces several critical best practices:

* GPU acceleration should be **selectively applied**, rather than used indiscriminately.
* Small and interactive FEM problems are better served by optimized CPU execution.
* For large-scale production workloads, **RawKernel-based GPU implementations provide the highest return on investment**.
* At scale, the **sparse linear solver—not the assembly kernel—becomes the dominant bottleneck**.

These results establish a clear upper bound for single-GPU FEM performance on the RTX 5090 within this study, validating the architectural choices adopted in the GPU implementations while providing quantitative evidence of when and why GPU acceleration is most effective. The RTX 5090 demonstrates **excellent scalability for medium to extreme FEM problem sizes**, delivering order-of-magnitude speedups when the problem scale justifies GPU usage. At the same time, the results clearly show that **hardware capability alone is insufficient**: algorithmic structure, solver behavior, and execution model ultimately determine performance.

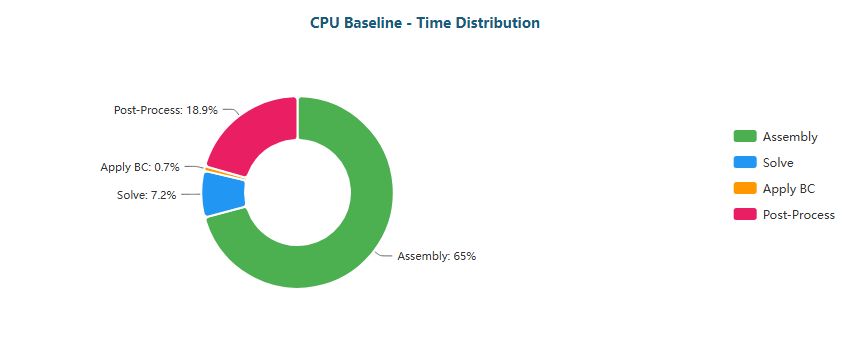
### 1.1.2 RTX 5090 Bottleneck Evolution Critical Analysis

As optimizations progress, the computational bottleneck shifts:

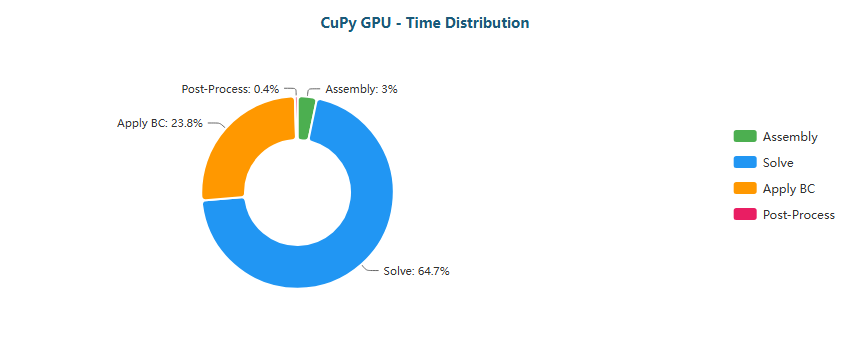
#### Backward-Facing Step (XS) - 287 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (65%) | Post-Proc (19%) |
| CPU Threaded | Assembly (48%) | Post-Proc (15%) |
| CPU Multiprocess | Post-Proc (62%) | Assembly (38%) |
| Numba CPU | Solve (51%) | Assembly (12%) |
| Numba CUDA | Solve (77%) | Assembly (11%) |
| CuPy GPU | Solve (65%) | BC (24%) |

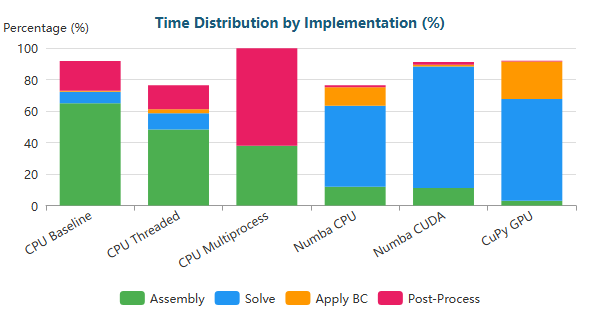
**Time Distribution:**



Pie



Pie

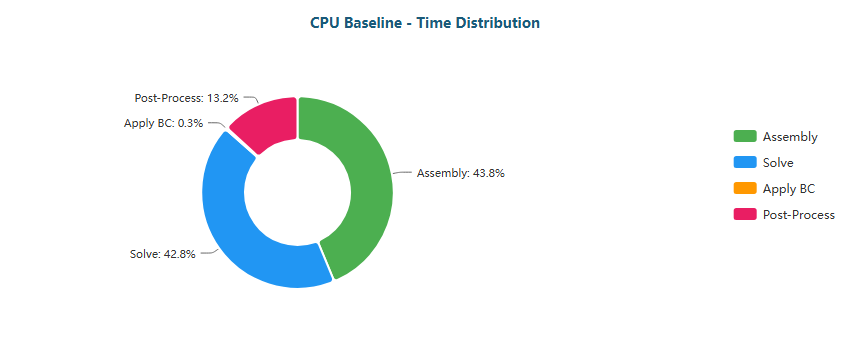


Stacked Bar

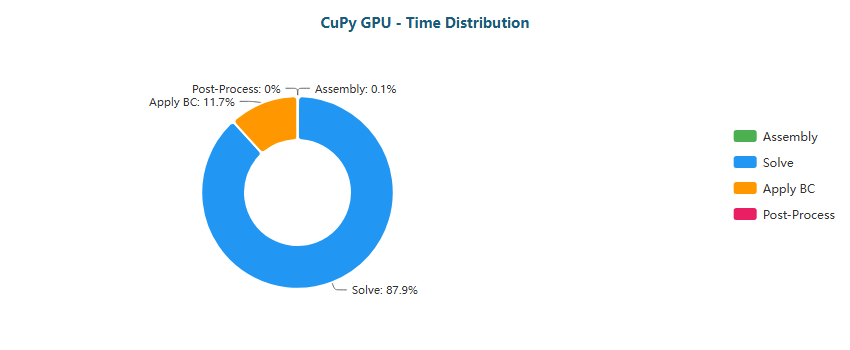
#### Backward-Facing Step (M) - 195,362 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (44%) | Solve (43%) |
| CPU Threaded | Solve (62%) | Assembly (25%) |
| CPU Multiprocess | Solve (53%) | BC (19%) |
| Numba CPU | Solve (97%) | BC (3%) |
| Numba CUDA | Solve (47%) | Assembly (30%) |
| CuPy GPU | Solve (88%) | BC (12%) |

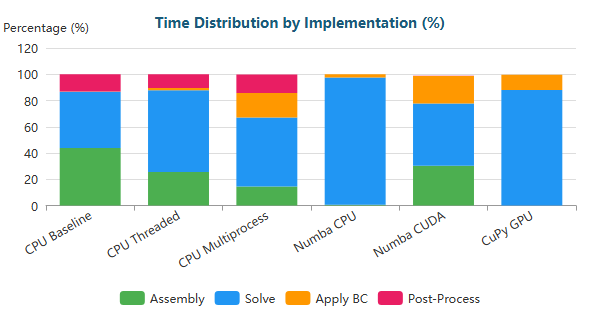
**Time Distribution:**



Pie



Pie

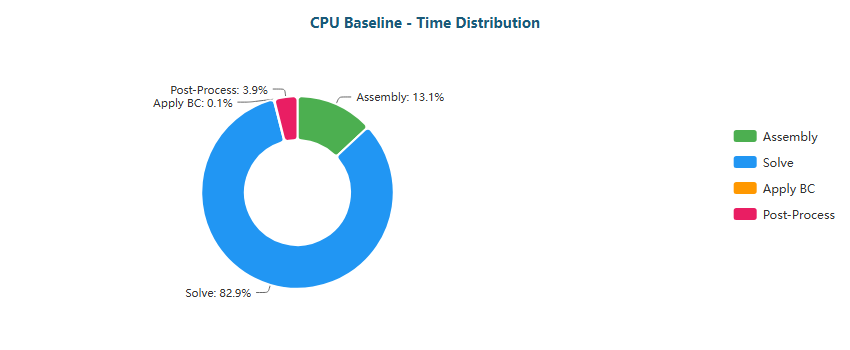


Stacked Bar

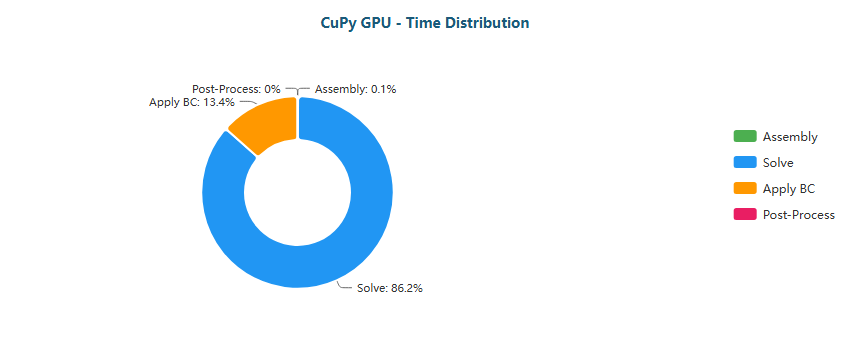
#### Backward-Facing Step (L) - 766,088 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (83%) | Assembly (13%) |
| CPU Threaded | Solve (79%) | Assembly (14%) |
| CPU Multiprocess | Solve (72%) | BC (22%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Assembly (36%) | Solve (36%) |
| CuPy GPU | Solve (86%) | BC (13%) |

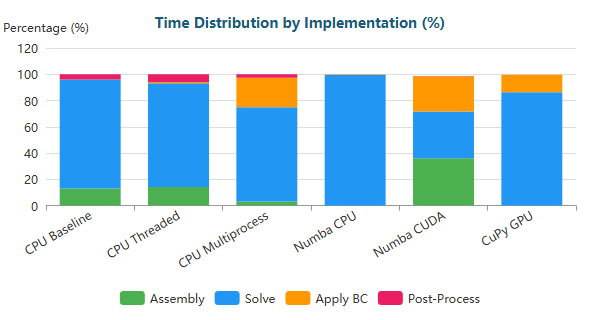
**Time Distribution:**



Pie



Pie



Stacked Bar

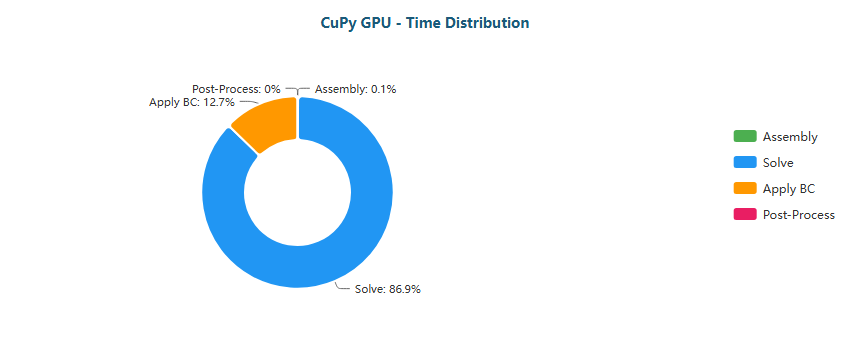
#### Backward-Facing Step (XL) - 1,283,215 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (76%) | Assembly (19%) |
| CPU Threaded | Solve (84%) | Assembly (11%) |
| CPU Multiprocess | Solve (75%) | BC (22%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Assembly (38%) | Solve (34%) |
| CuPy GPU | Solve (87%) | BC (13%) |

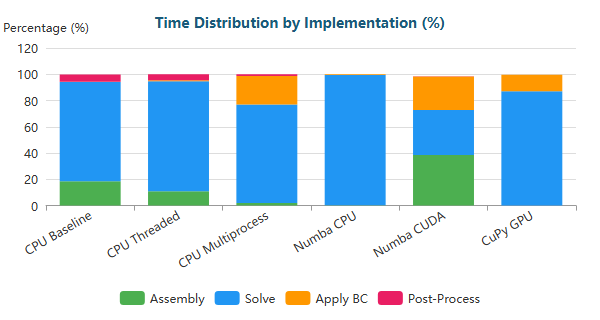
**Time Distribution:**



Pie



Pie

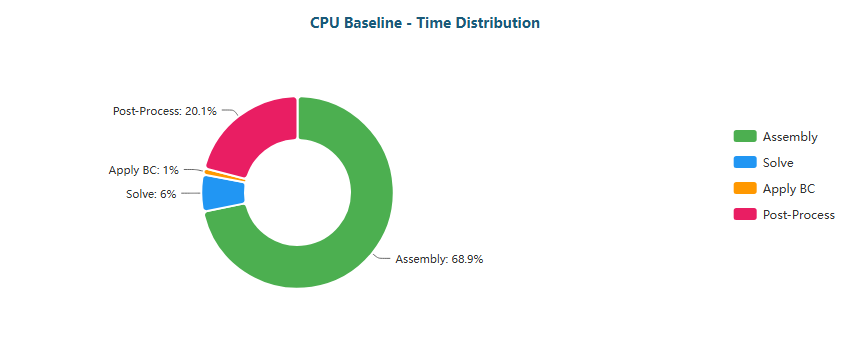


Stacked Bar

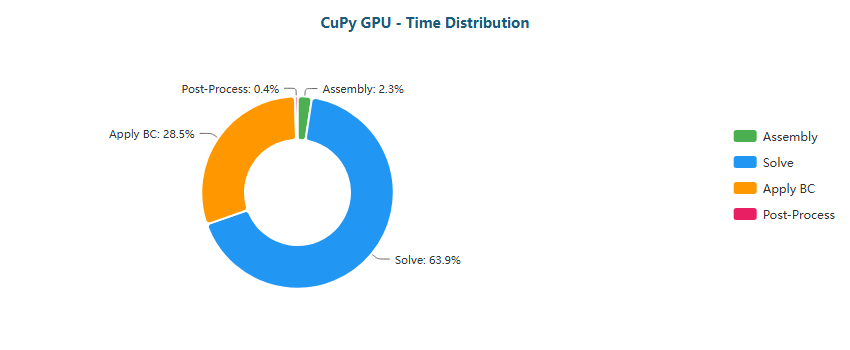
#### Elbow 90° (XS) - 411 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (69%) | Post-Proc (20%) |
| CPU Threaded | Assembly (51%) | Post-Proc (19%) |
| CPU Multiprocess | Post-Proc (50%) | Assembly (50%) |
| Numba CPU | Solve (49%) | BC (17%) |
| Numba CUDA | Solve (80%) | Assembly (9%) |
| CuPy GPU | Solve (64%) | BC (28%) |

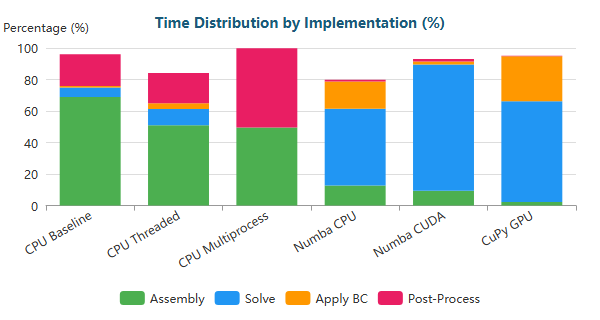
**Time Distribution:**



Pie



Pie

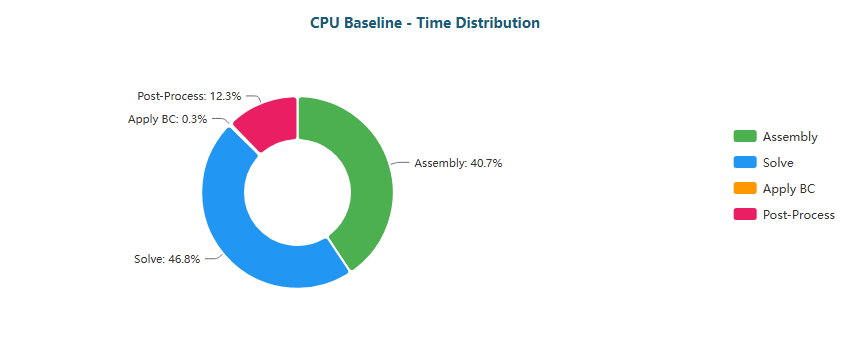


Stacked Bar

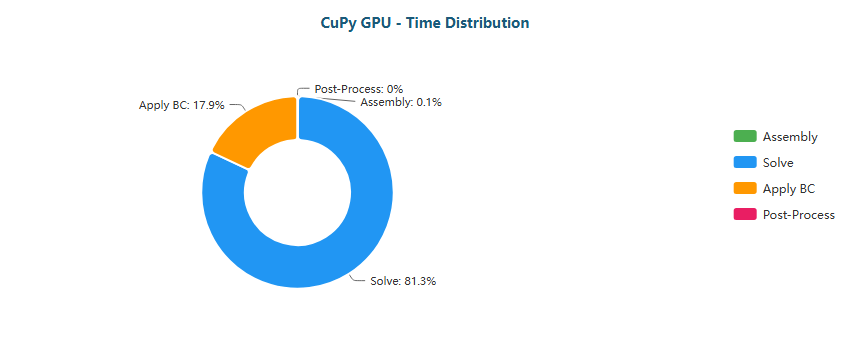
#### Elbow 90° (M) - 161,984 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (47%) | Assembly (41%) |
| CPU Threaded | Solve (58%) | Assembly (29%) |
| CPU Multiprocess | Solve (35%) | BC (35%) |
| Numba CPU | Solve (97%) | BC (2%) |
| Numba CUDA | Solve (47%) | Assembly (31%) |
| CuPy GPU | Solve (81%) | BC (18%) |

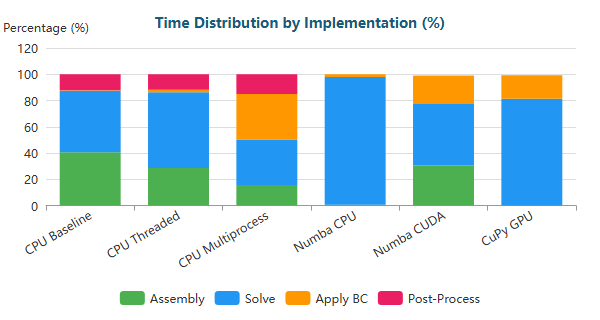
**Time Distribution:**



Pie



Pie

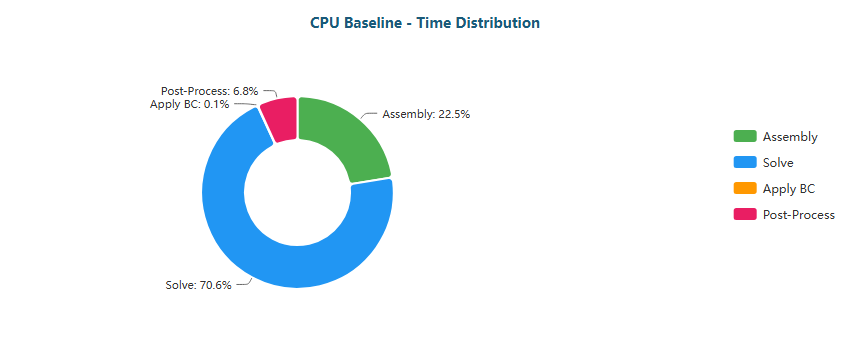


Stacked Bar

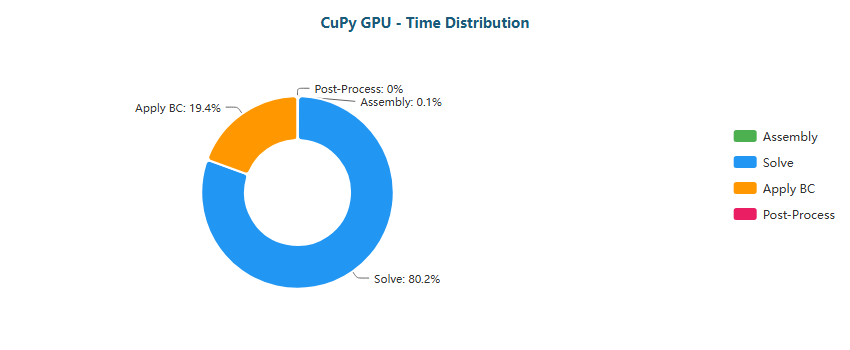
#### Elbow 90° (L) - 623,153 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (71%) | Assembly (22%) |
| CPU Threaded | Solve (74%) | Assembly (18%) |
| CPU Multiprocess | Solve (49%) | BC (44%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Assembly (40%) | Solve (32%) |
| CuPy GPU | Solve (80%) | BC (19%) |

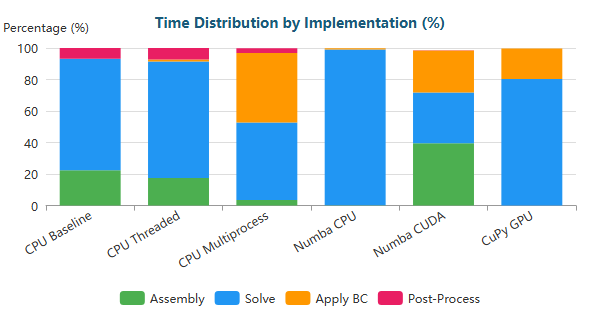
**Time Distribution:**



Pie



Pie

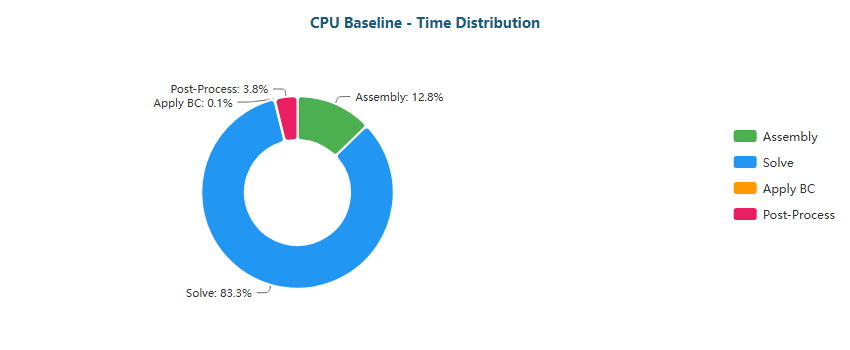


Stacked Bar

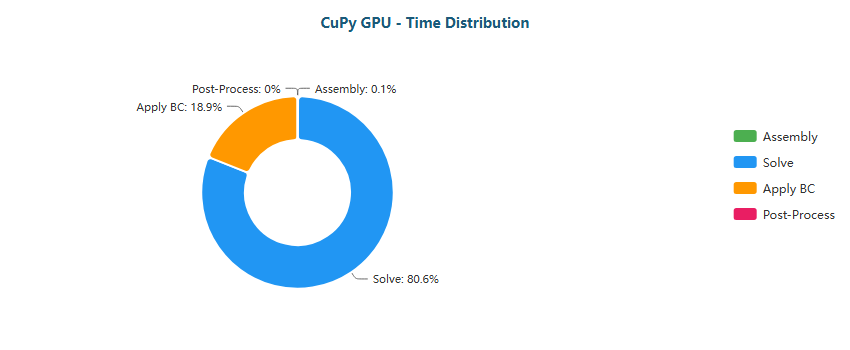
#### Elbow 90° (XL) - 1,044,857 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (83%) | Assembly (13%) |
| CPU Threaded | Solve (79%) | Assembly (14%) |
| CPU Multiprocess | Solve (51%) | BC (45%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Assembly (37%) | Solve (35%) |
| CuPy GPU | Solve (81%) | BC (19%) |

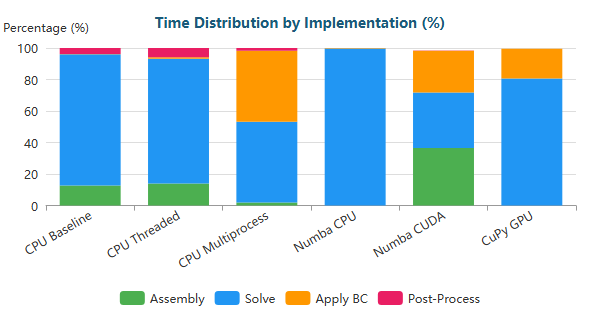
**Time Distribution:**



Pie



Pie

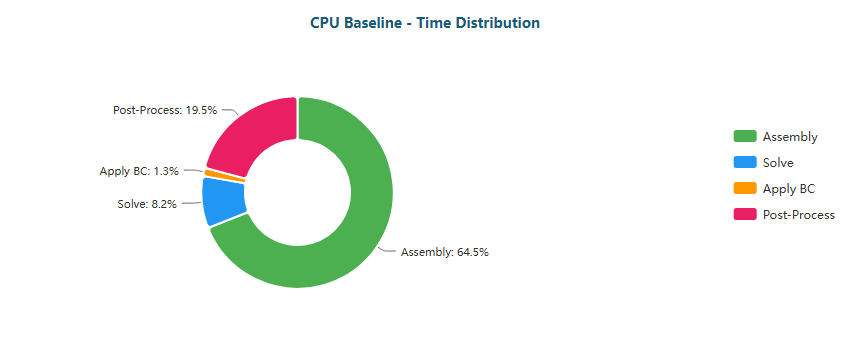


Stacked Bar

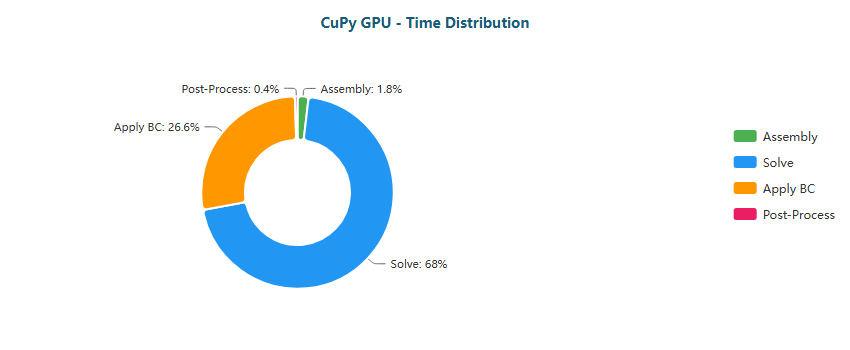
#### S-Bend (XS) - 387 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (65%) | Post-Proc (19%) |
| CPU Threaded | Assembly (50%) | Post-Proc (17%) |
| CPU Multiprocess | Post-Proc (62%) | Assembly (38%) |
| Numba CPU | Solve (52%) | BC (16%) |
| Numba CUDA | Solve (81%) | Assembly (10%) |
| CuPy GPU | Solve (68%) | BC (27%) |

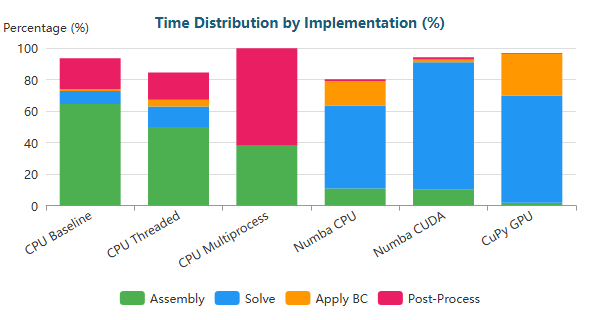
**Time Distribution:**



Pie



Pie



Stacked Bar

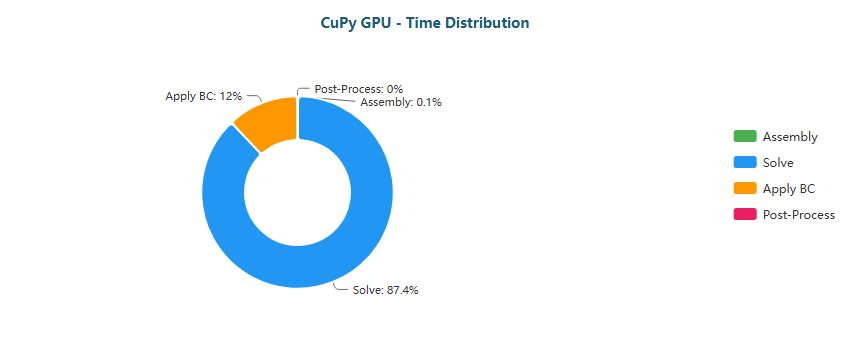
#### S-Bend (M) - 196,078 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (47%) | Assembly (40%) |
| CPU Threaded | Solve (66%) | Assembly (23%) |
| CPU Multiprocess | Solve (60%) | Assembly (14%) |
| Numba CPU | Solve (97%) | BC (2%) |
| Numba CUDA | Solve (59%) | Assembly (23%) |
| CuPy GPU | Solve (87%) | BC (12%) |

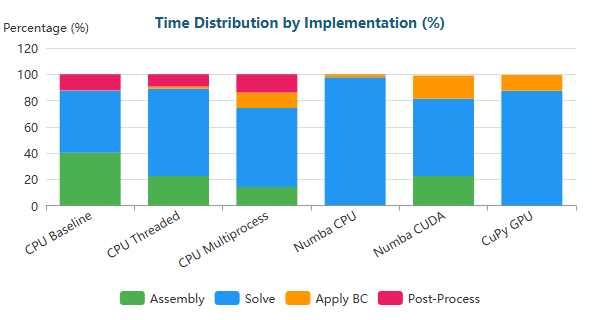
**Time Distribution:**



Pie



Pie

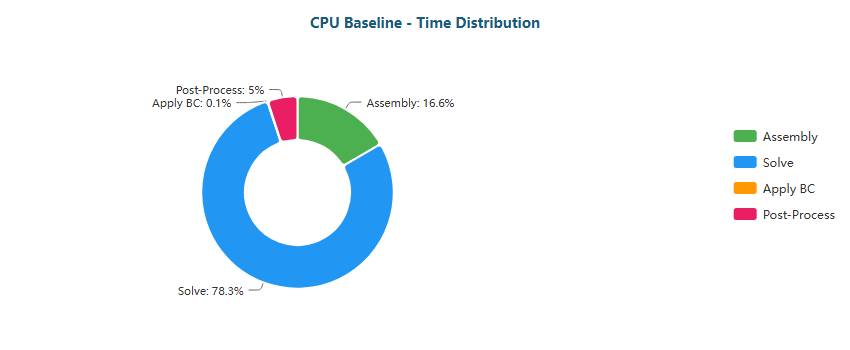


Stacked Bar

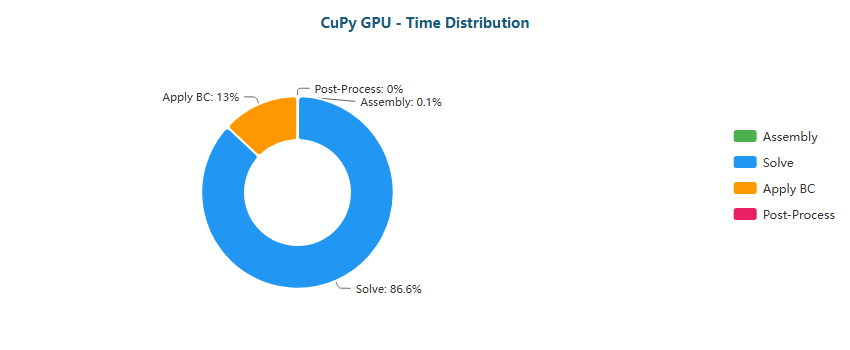
#### S-Bend (L) - 765,441 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (78%) | Assembly (17%) |
| CPU Threaded | Solve (81%) | Assembly (13%) |
| CPU Multiprocess | Solve (80%) | BC (15%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (39%) | Assembly (36%) |
| CuPy GPU | Solve (87%) | BC (13%) |

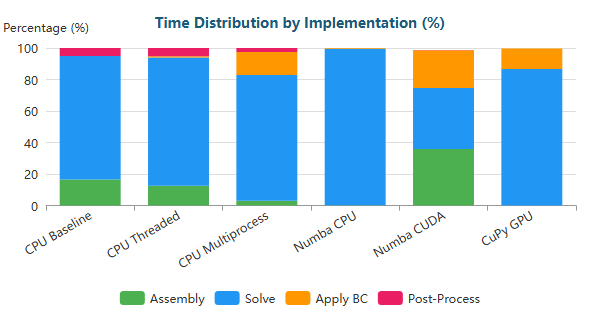
**Time Distribution:**



Pie



Pie

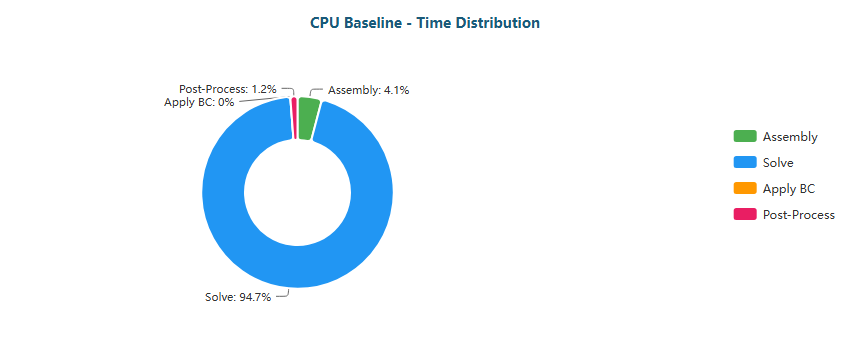


Stacked Bar

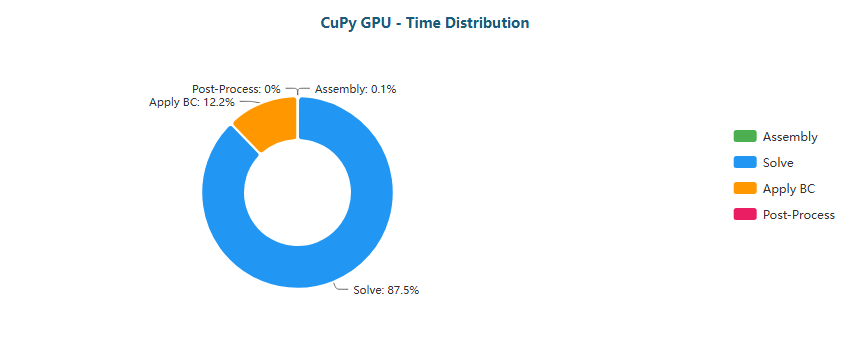
#### S-Bend (XL) - 1,286,039 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (95%) | Assembly (4%) |
| CPU Threaded | Solve (85%) | Assembly (10%) |
| CPU Multiprocess | Solve (83%) | BC (14%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (39%) | Assembly (36%) |
| CuPy GPU | Solve (87%) | BC (12%) |

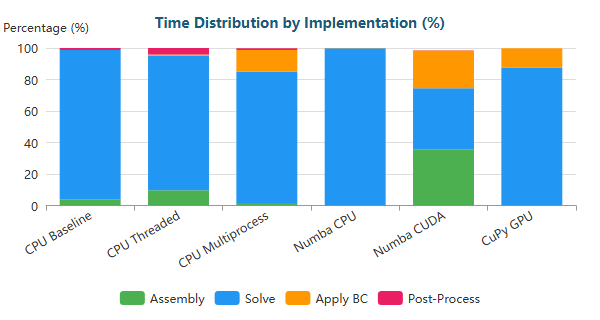
**Time Distribution:**



Pie



Pie

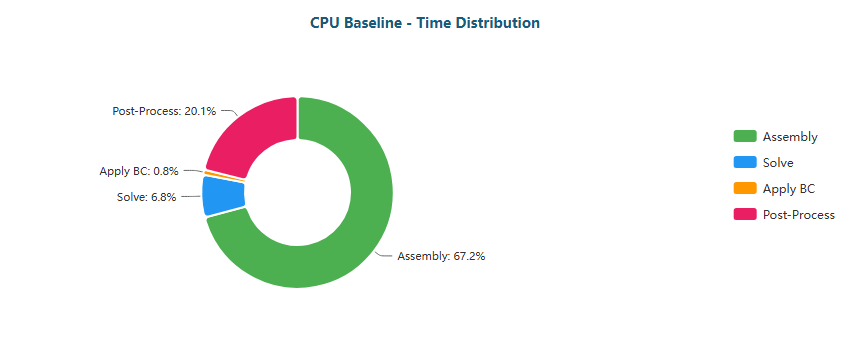


Stacked Bar

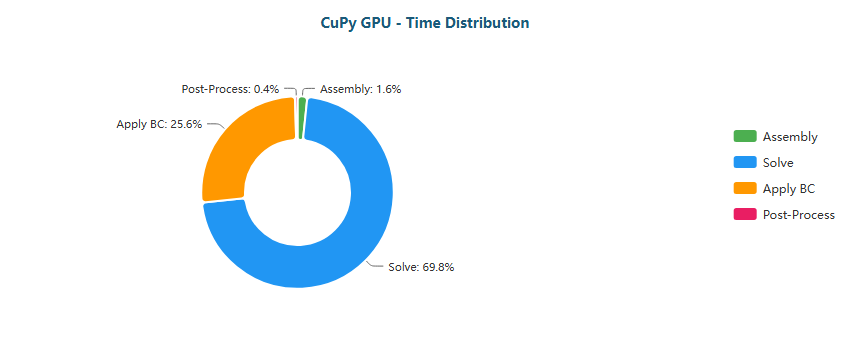
#### T-Junction (XS) - 393 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (67%) | Post-Proc (20%) |
| CPU Threaded | Assembly (51%) | Post-Proc (19%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (52%) | BC (15%) |
| Numba CUDA | Solve (85%) | Assembly (8%) |
| CuPy GPU | Solve (70%) | BC (26%) |

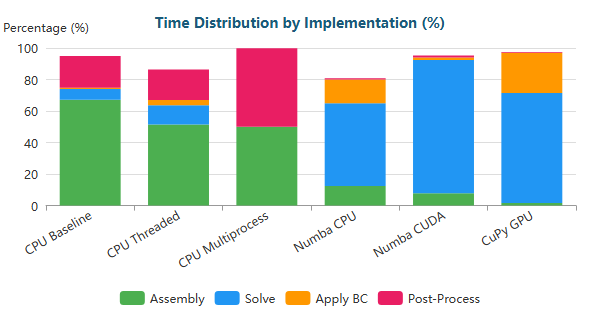
**Time Distribution:**



Pie



Pie

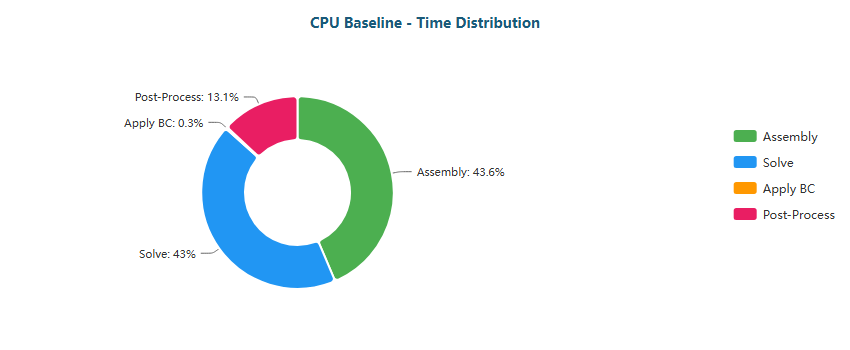


Stacked Bar

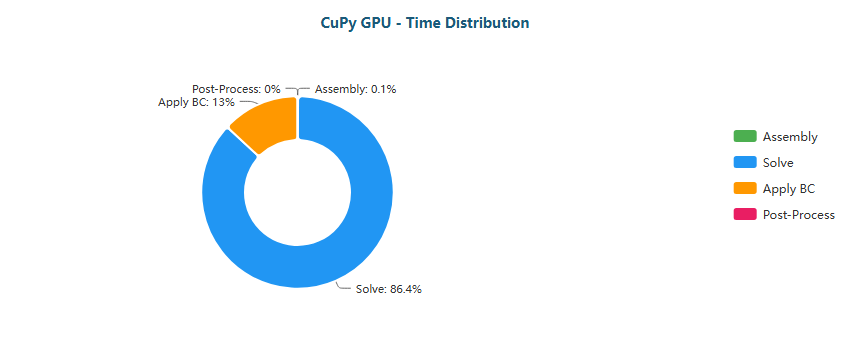
#### T-Junction (M) - 196,420 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (44%) | Solve (43%) |
| CPU Threaded | Solve (64%) | Assembly (24%) |
| CPU Multiprocess | Solve (58%) | Assembly (15%) |
| Numba CPU | Solve (97%) | BC (2%) |
| Numba CUDA | Solve (49%) | Assembly (29%) |
| CuPy GPU | Solve (86%) | BC (13%) |

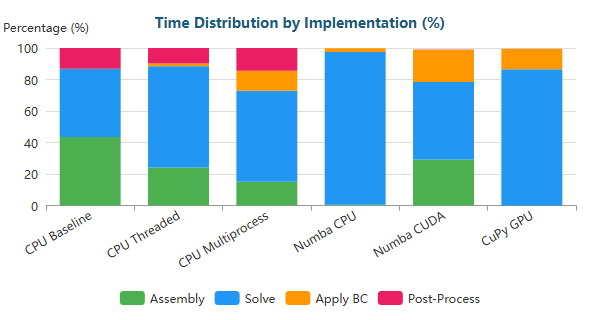
**Time Distribution:**



Pie



Pie

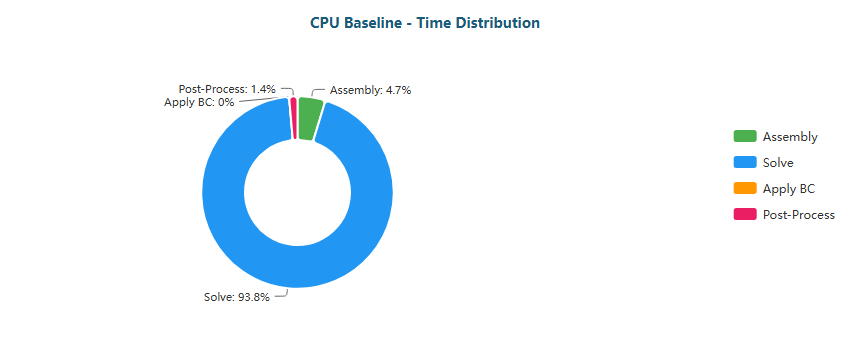


Stacked Bar

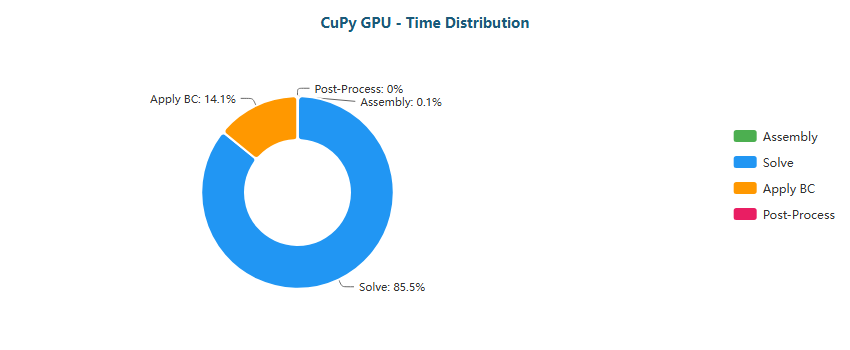
#### T-Junction (L) - 768,898 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (94%) | Assembly (5%) |
| CPU Threaded | Solve (80%) | Assembly (13%) |
| CPU Multiprocess | Solve (79%) | BC (15%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (37%) | Assembly (35%) |
| CuPy GPU | Solve (85%) | BC (14%) |

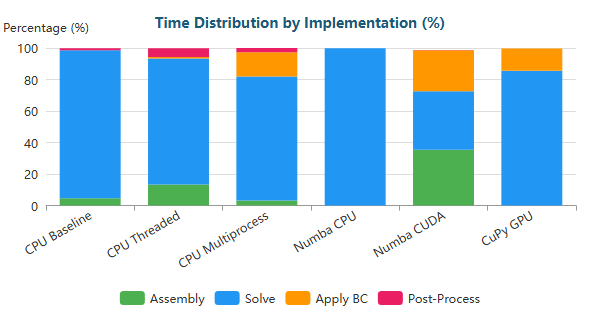
**Time Distribution:**



Pie



Pie

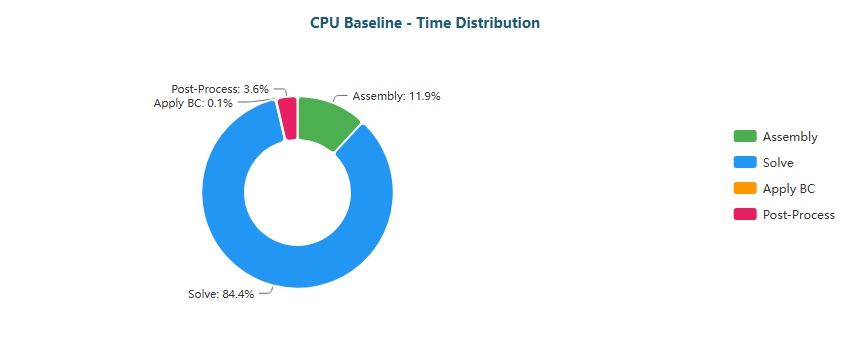


Stacked Bar

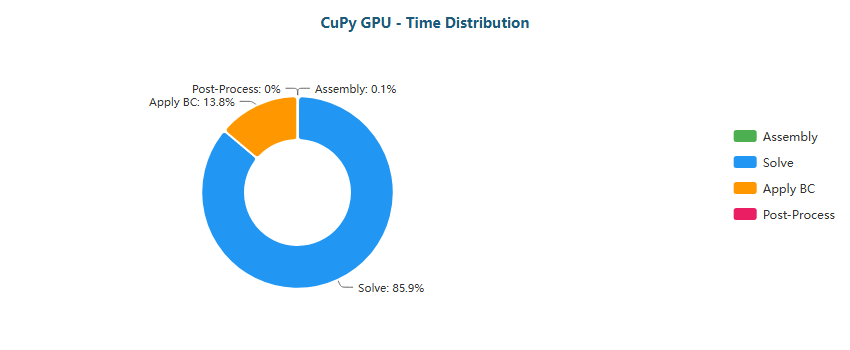
#### T-Junction (XL) - 1,291,289 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (84%) | Assembly (12%) |
| CPU Threaded | Solve (84%) | Assembly (10%) |
| CPU Multiprocess | Solve (82%) | BC (15%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Assembly (37%) | Solve (36%) |
| CuPy GPU | Solve (86%) | BC (14%) |

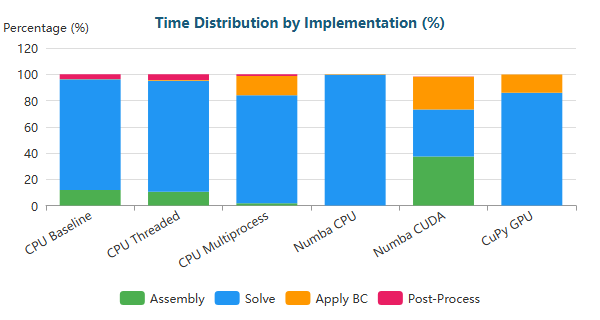
**Time Distribution:**



Pie



Pie

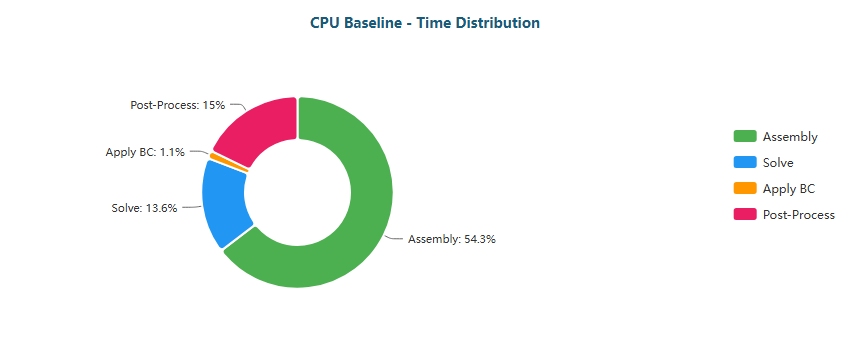


Stacked Bar

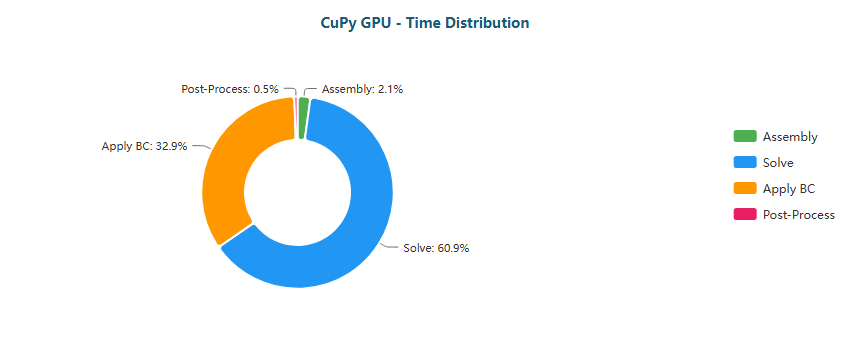
#### Venturi (XS) - 341 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (54%) | Post-Proc (15%) |
| CPU Threaded | Assembly (48%) | Post-Proc (19%) |
| CPU Multiprocess | Post-Proc (62%) | Assembly (37%) |
| Numba CPU | Solve (48%) | BC (16%) |
| Numba CUDA | Solve (78%) | Assembly (10%) |
| CuPy GPU | Solve (61%) | BC (33%) |

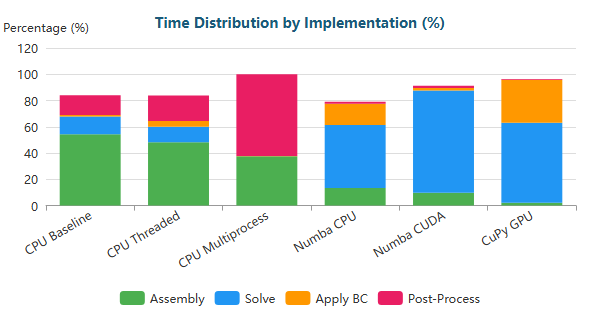
**Time Distribution:**



Pie



Pie

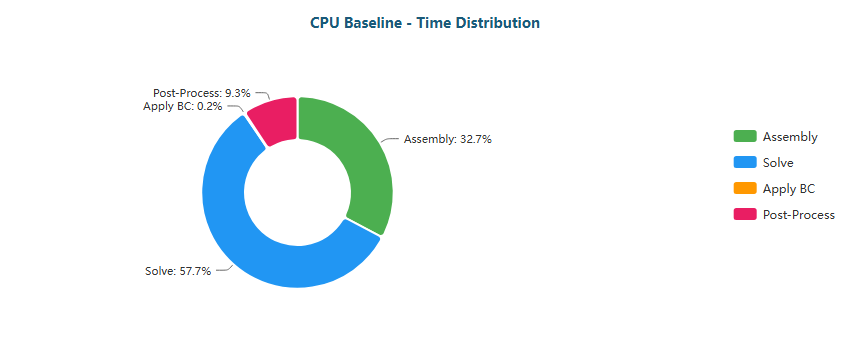


Stacked Bar

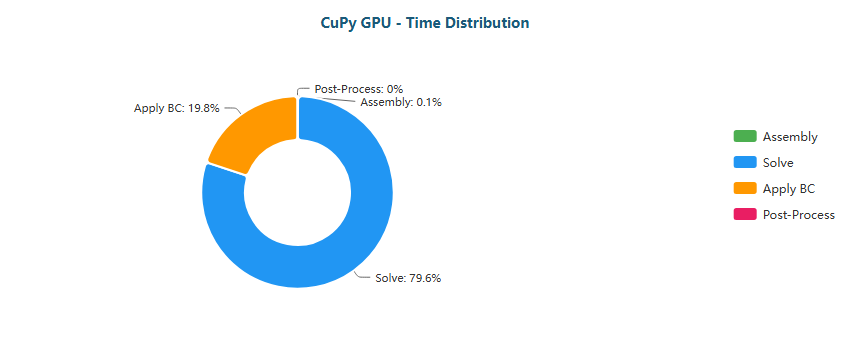
#### Venturi (M) - 194,325 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (58%) | Assembly (33%) |
| CPU Threaded | Solve (63%) | Assembly (25%) |
| CPU Multiprocess | Solve (45%) | BC (22%) |
| Numba CPU | Solve (97%) | BC (3%) |
| Numba CUDA | Solve (54%) | Assembly (26%) |
| CuPy GPU | Solve (80%) | BC (20%) |

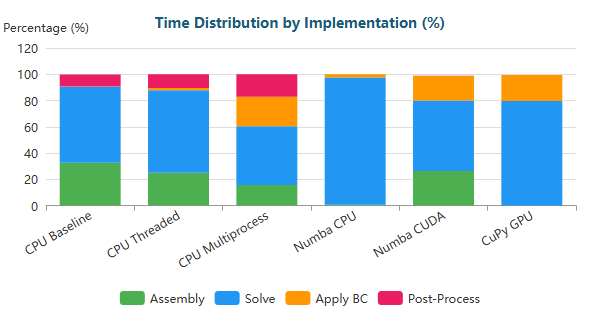
**Time Distribution:**



Pie



Pie

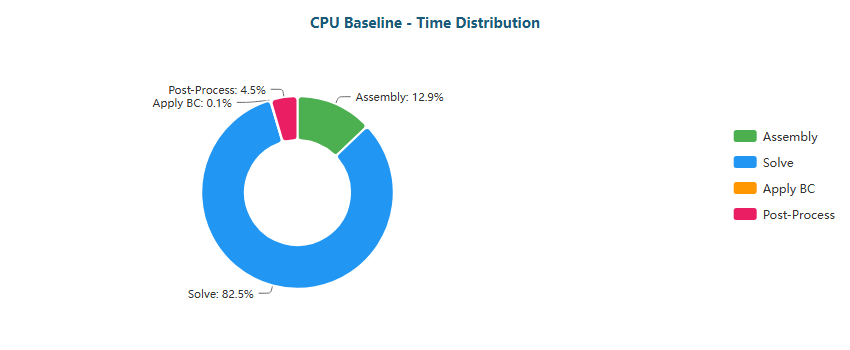


Stacked Bar

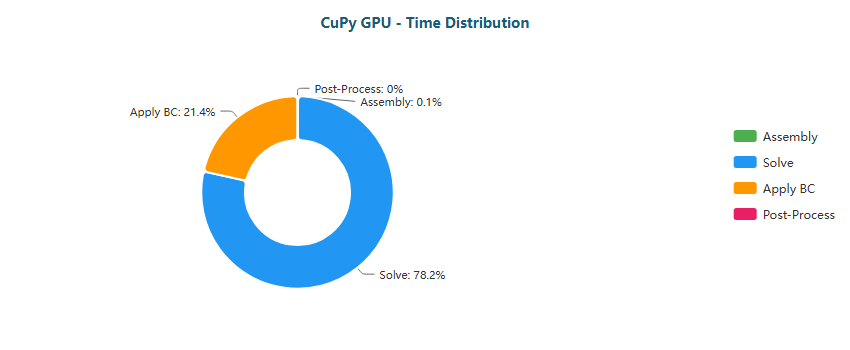
#### Venturi (L) - 763,707 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (83%) | Assembly (13%) |
| CPU Threaded | Solve (78%) | Assembly (15%) |
| CPU Multiprocess | Solve (65%) | BC (28%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (40%) | Assembly (32%) |
| CuPy GPU | Solve (78%) | BC (21%) |

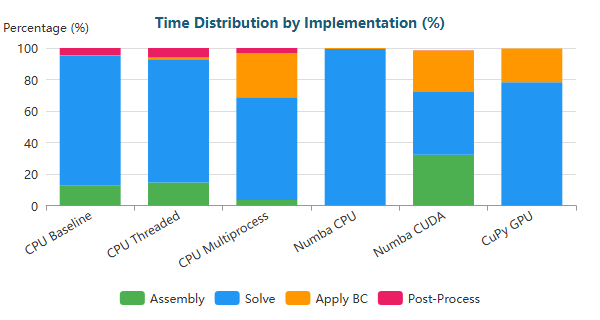
**Time Distribution:**



Pie



Pie

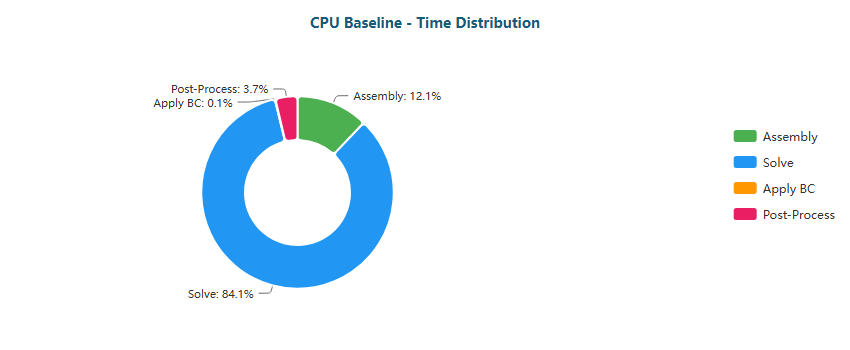


Stacked Bar

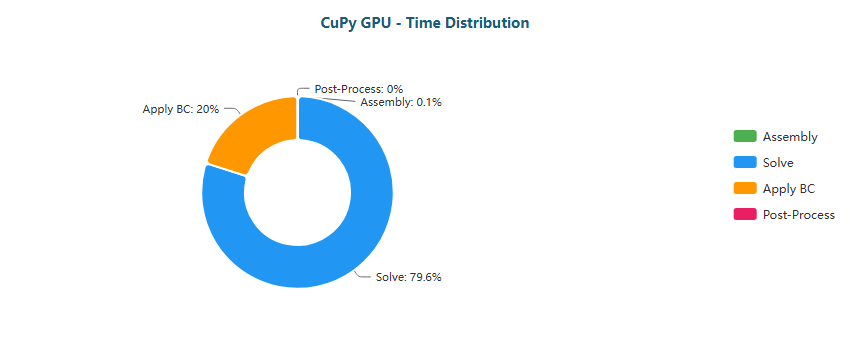
#### Venturi (XL) - 1,284,412 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (84%) | Assembly (12%) |
| CPU Threaded | Solve (83%) | Assembly (12%) |
| CPU Multiprocess | Solve (68%) | BC (29%) |
| Numba CPU | Solve (99%) | BC (0%) |
| Numba CUDA | Solve (38%) | Assembly (33%) |
| CuPy GPU | Solve (80%) | BC (20%) |

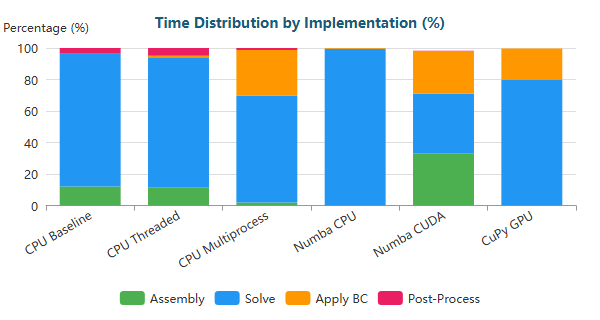
**Time Distribution:**



Pie



Pie

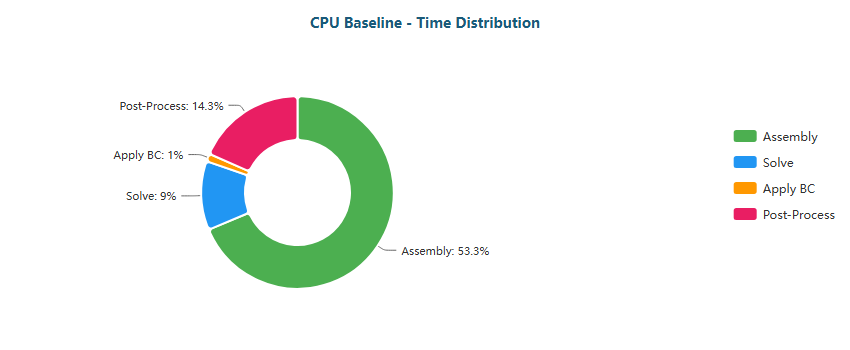


Stacked Bar

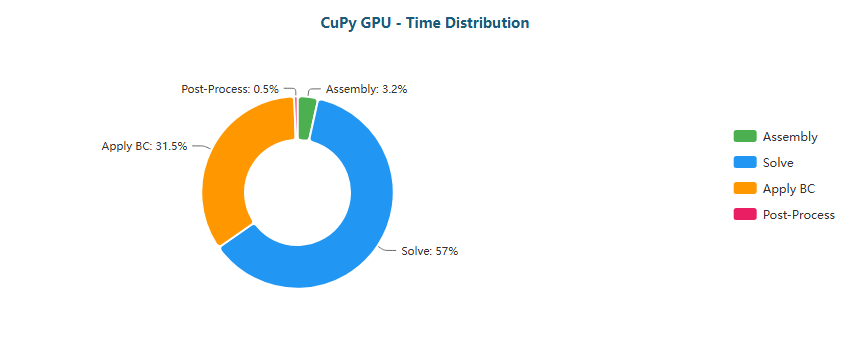
#### Y-Shaped (XS) - 201 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (53%) | Post-Proc (14%) |
| CPU Threaded | Assembly (42%) | Post-Proc (13%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (48%) | Assembly (13%) |
| Numba CUDA | Solve (73%) | Assembly (12%) |
| CuPy GPU | Solve (57%) | BC (31%) |

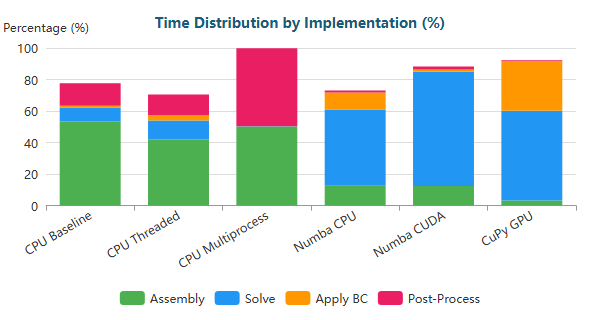
**Time Distribution:**



Pie



Pie



Stacked Bar

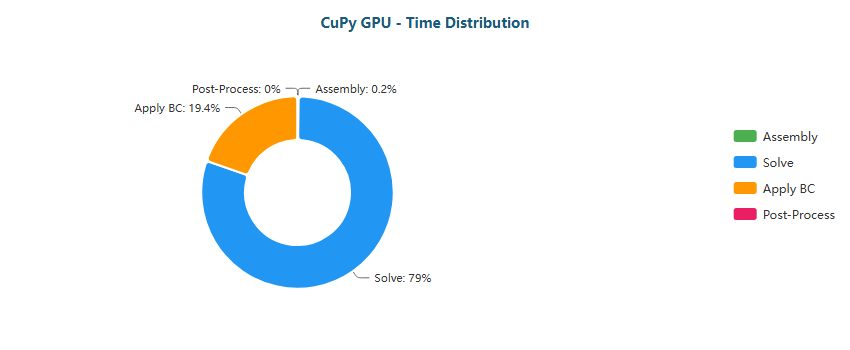
#### Y-Shaped (M) - 195,853 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (55%) | Assembly (33%) |
| CPU Threaded | Solve (66%) | Assembly (23%) |
| CPU Multiprocess | Solve (47%) | Assembly (22%) |
| Numba CPU | Solve (96%) | BC (3%) |
| Numba CUDA | Solve (47%) | Assembly (26%) |
| CuPy GPU | Solve (79%) | BC (19%) |

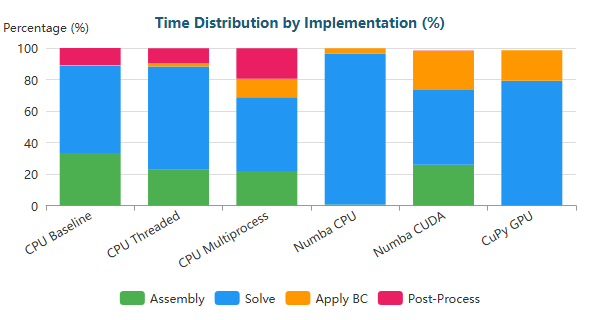
**Time Distribution:**



Pie



Pie

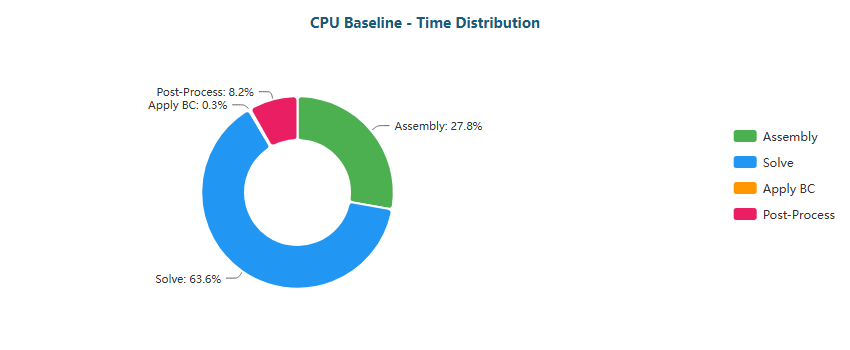


Stacked Bar

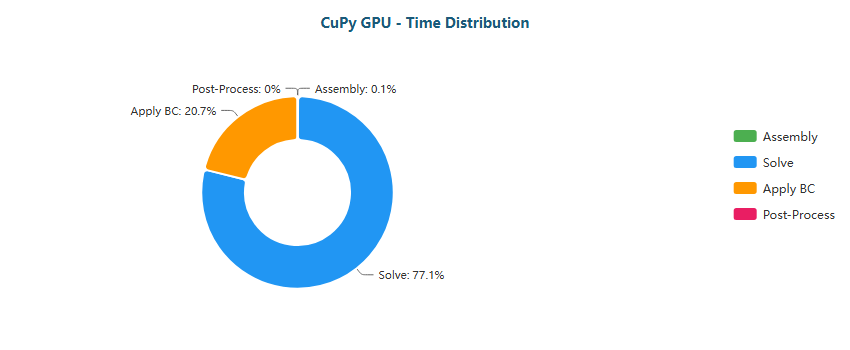
#### Y-Shaped (L) - 772,069 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (64%) | Assembly (28%) |
| CPU Threaded | Solve (81%) | Assembly (13%) |
| CPU Multiprocess | Solve (76%) | BC (16%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (34%) | Assembly (33%) |
| CuPy GPU | Solve (77%) | BC (21%) |

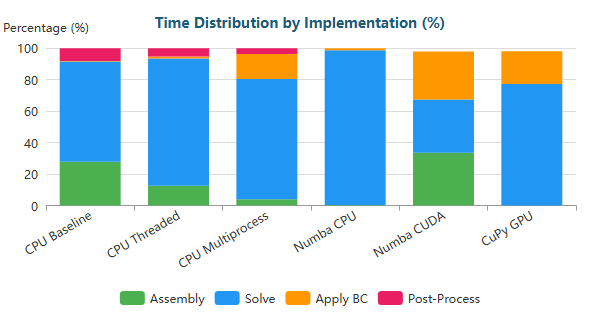
**Time Distribution:**



Pie



Pie

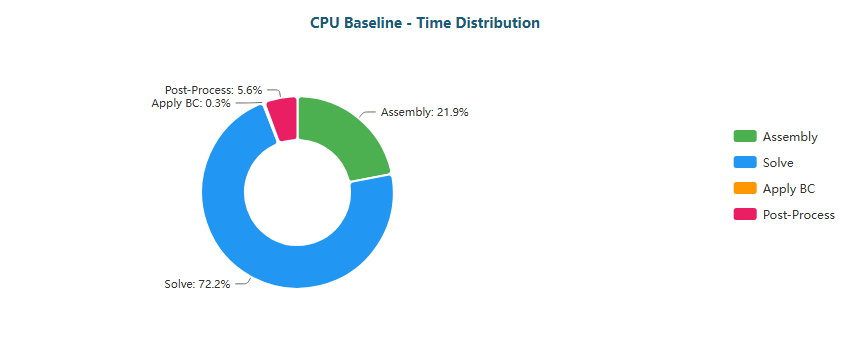


Stacked Bar

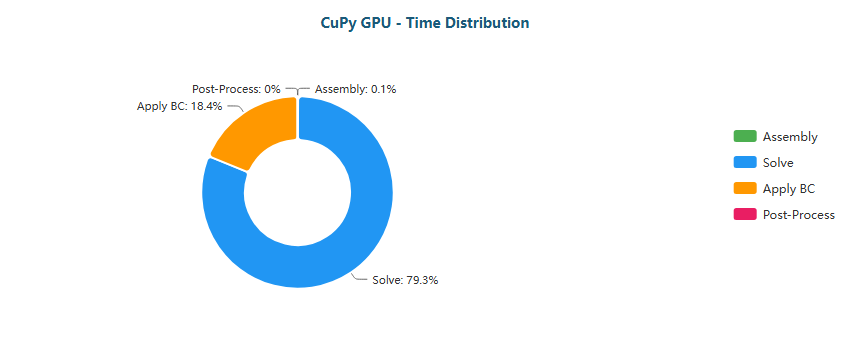
#### Y-Shaped (XL) - 1,357,953 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (72%) | Assembly (22%) |
| CPU Threaded | Solve (86%) | Assembly (9%) |
| CPU Multiprocess | Solve (81%) | BC (16%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Assembly (33%) | BC (32%) |
| CuPy GPU | Solve (79%) | BC (18%) |

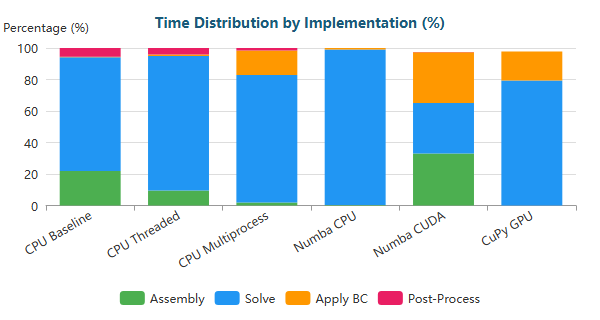
**Time Distribution:**



Pie



Pie



Stacked Bar

### Why Each Optimization Helps

| Transition | Reason |
| --- | --- |
| Baseline → Threaded | Limited by Python GIL; threads only help for I/O |
| Threaded → Multiprocess | Bypasses GIL via separate processes; IPC overhead limits gains |
| Multiprocess → Numba CPU | JIT compilation eliminates interpreter overhead; true parallel loops |
| Numba CPU → Numba CUDA | GPU parallelism: thousands of threads vs dozens of CPU cores |
| Numba CUDA → CuPy GPU | CUDA C kernels more optimized than Numba-generated PTX |

#### 1.1.2.1 Bottleneck Migration on RTX 5090

From a bottleneck-analysis perspective, the results on the RTX 5090 reveal a clear and systematic migration of performance constraints as optimization levels increase:

* **CPU baseline and threaded executions:** Dominated by *Assembly* and *Post-Processing*, especially in XS meshes, reflecting Python overhead and limited parallel efficiency.
* **Multiprocessing:** Reduces assembly cost but introduces heavy *Post-Processing* and IPC overheads, preventing scalable gains.
* **Numba JIT CPU:** Successfully removes interpreter overhead, shifting the bottleneck almost entirely to the *Solve* phase, particularly for M, L, and XL meshes.
* **GPU-based executions (Numba CUDA and CuPy):** Assembly and post-processing become negligible (<2–3% in most M/L/XL cases), fully exposing the *linear solver* as the dominant bottleneck.
* **CuPy GPU (RawKernel):** Consistently shows *Solve* accounting for ~80–90% of total runtime, confirming that sparse linear algebra—not kernel execution—is the limiting factor.

These results demonstrate that the RTX 5090 does not shift the bottleneck back to computation, but rather exposes the **memory-bound nature of sparse solvers**, particularly SpMV operations within the Conjugate Gradient method.

#### 1.1.2.2 Optimization Implications and Performance Limits

From an optimization standpoint, the bottleneck behavior on the RTX 5090 implies the following practical conclusions:

* **Further assembly optimization yields diminishing returns**, as this stage is already effectively eliminated in GPU executions.
* **Solver efficiency is the primary performance lever**:
  + Reducing iteration counts via better preconditioning.
  + Improving sparse matrix layout and access locality.
* **Boundary condition application emerges as a secondary bottleneck**, especially in XS meshes, due to kernel-launch overheads and synchronization costs.
* **GPU acceleration is scale-dependent**:
  + **XS meshes:** CPU JIT execution remains preferable due to lower fixed overhead.
  + **M, L, XL meshes:** GPU execution is mandatory, but gains are capped by solver memory behavior.
* **Hardware capability alone is insufficient**: the RTX 5090 exposes algorithmic limits rather than removing them.

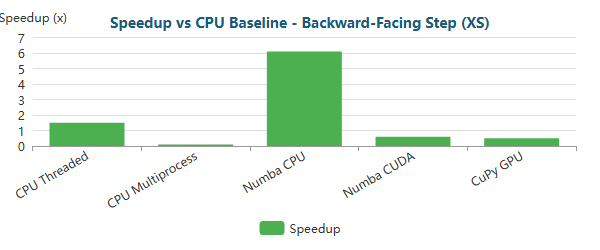
The RTX 5090 marks a transition point where FEM performance is constrained not by raw compute power, but by **algorithmic structure, memory access patterns, and solver design**, establishing a clear upper bound for single-GPU performance in this study.

### 1.2 RTX 4090 Performance

Key results from performance benchmarks comparing FEM solver implementations.

**Backward-Facing Step (XS)** (287 nodes)

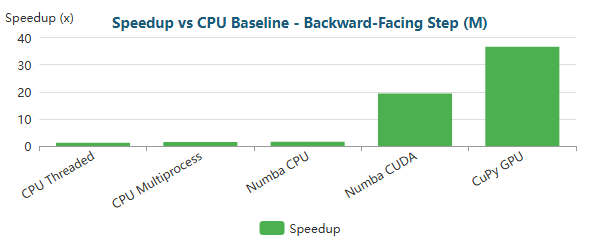
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 25ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 17ms ± 2ms | 1.5x | 3 |
| CPU Multiprocess | 345ms ± 38ms | 0.1x | 3 |
| Numba CPU | <0.01s ± 0ms | 6.1x | 3 |
| Numba CUDA | 45ms ± 10ms | 0.6x | 3 |
| CuPy GPU | 51ms ± 4ms | 0.5x | 3 |



Bar

**Backward-Facing Step (M)** (195,362 nodes)

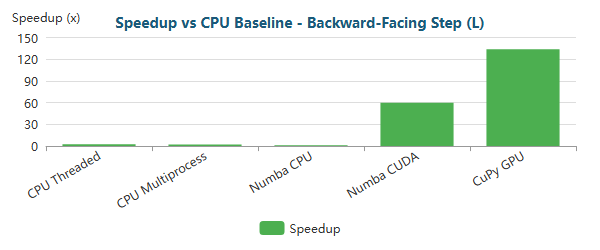
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 46.68s ± 1.66s | 1.0x | 3 |
| CPU Threaded | 37.41s ± 0.38s | 1.2x | 3 |
| CPU Multiprocess | 32.18s ± 1.61s | 1.5x | 3 |
| Numba CPU | 29.17s ± 2.04s | 1.6x | 3 |
| Numba CUDA | 2.40s ± 0.03s | 19.4x | 3 |
| CuPy GPU | 1.27s ± 0.02s | 36.6x | 3 |



Bar

**Backward-Facing Step (L)** (766,088 nodes)

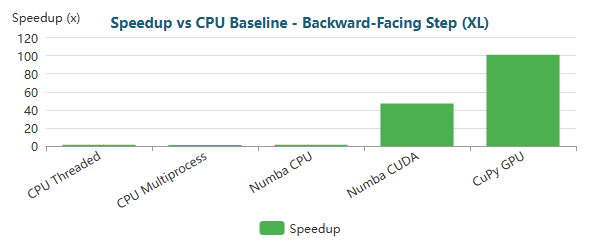
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 8m 2.0s ± 1.2s | 1.0x | 3 |
| CPU Threaded | 3m 2.0s ± 2.3s | 2.6x | 3 |
| CPU Multiprocess | 3m 34.5s ± 2.4s | 2.2x | 3 |
| Numba CPU | 6m 15.5s ± 0.8s | 1.3x | 3 |
| Numba CUDA | 8.07s ± 0.08s | 59.7x | 3 |
| CuPy GPU | 3.61s ± 0.03s | 133.6x | 3 |



Bar

**Backward-Facing Step (XL)** (1,283,215 nodes)

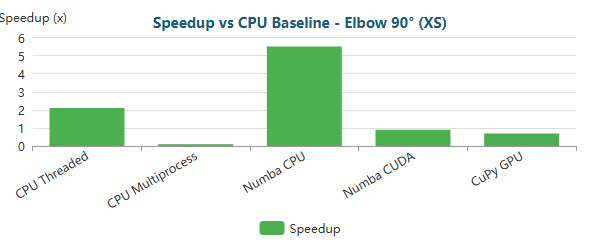
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 11m 3.6s ± 7.2s | 1.0x | 3 |
| CPU Threaded | 6m 55.3s ± 3.6s | 1.6x | 3 |
| CPU Multiprocess | 8m 21.9s ± 10.9s | 1.3x | 3 |
| Numba CPU | 8m 2.4s ± 7.0s | 1.4x | 3 |
| Numba CUDA | 14.05s ± 0.14s | 47.2x | 3 |
| CuPy GPU | 6.57s ± 0.01s | 101.0x | 3 |



Bar

**Elbow 90° (XS)** (411 nodes)

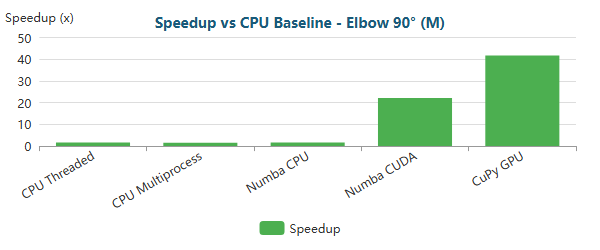
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 38ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 18ms ± 1ms | 2.1x | 3 |
| CPU Multiprocess | 374ms ± 42ms | 0.1x | 3 |
| Numba CPU | <0.01s ± 2ms | 5.5x | 3 |
| Numba CUDA | 44ms ± 1ms | 0.9x | 3 |
| CuPy GPU | 53ms ± 2ms | 0.7x | 3 |



Bar

**Elbow 90° (M)** (161,984 nodes)

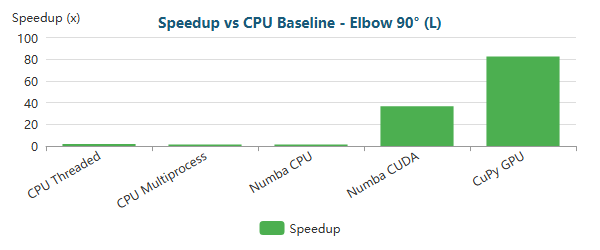
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 43.70s ± 0.62s | 1.0x | 3 |
| CPU Threaded | 27.97s ± 0.44s | 1.6x | 3 |
| CPU Multiprocess | 28.68s ± 1.08s | 1.5x | 3 |
| Numba CPU | 27.40s ± 1.48s | 1.6x | 3 |
| Numba CUDA | 1.97s ± 0.02s | 22.1x | 3 |
| CuPy GPU | 1.05s ± 0.05s | 41.7x | 3 |



Bar

**Elbow 90° (L)** (623,153 nodes)

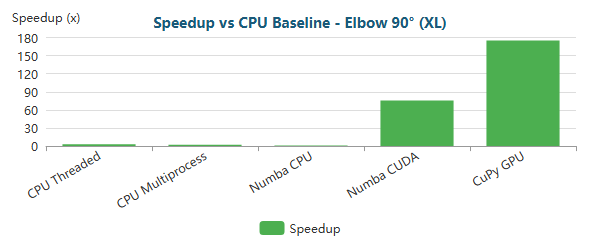
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 3m 39.3s ± 0.8s | 1.0x | 3 |
| CPU Threaded | 1m 58.3s ± 0.6s | 1.9x | 3 |
| CPU Multiprocess | 2m 34.8s ± 18.6s | 1.4x | 3 |
| Numba CPU | 2m 32.1s ± 1.3s | 1.4x | 3 |
| Numba CUDA | 5.98s ± 0.05s | 36.7x | 3 |
| CuPy GPU | 2.66s ± 0.02s | 82.4x | 3 |



Bar

**Elbow 90° (XL)** (1,044,857 nodes)

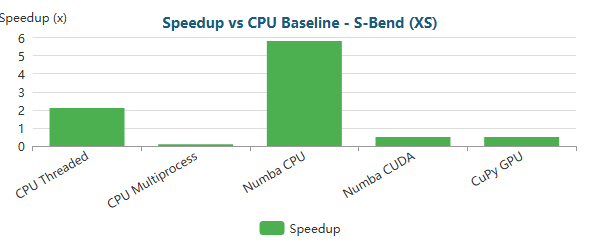
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 12m 54.9s ± 5.3s | 1.0x | 3 |
| CPU Threaded | 4m 19.0s ± 2.5s | 3.0x | 3 |
| CPU Multiprocess | 5m 30.3s ± 6.5s | 2.3x | 3 |
| Numba CPU | 11m 23.8s ± 6.0s | 1.1x | 3 |
| Numba CUDA | 10.27s ± 0.02s | 75.4x | 3 |
| CuPy GPU | 4.43s ± 0.04s | 175.0x | 3 |



Bar

**S-Bend (XS)** (387 nodes)

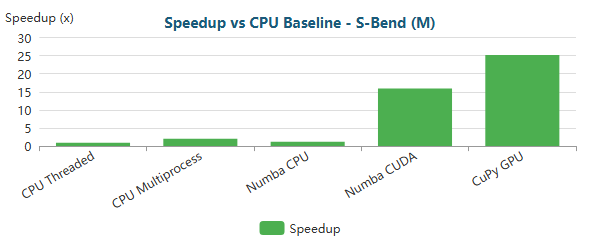
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 34ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 17ms ± 3ms | 2.1x | 3 |
| CPU Multiprocess | 320ms ± 15ms | 0.1x | 3 |
| Numba CPU | <0.01s ± 2ms | 5.8x | 3 |
| Numba CUDA | 63ms ± 6ms | 0.5x | 3 |
| CuPy GPU | 62ms ± 8ms | 0.5x | 3 |



Bar

**S-Bend (M)** (196,078 nodes)

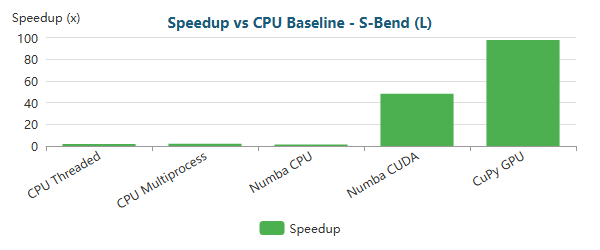
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 38.82s ± 0.42s | 1.0x | 3 |
| CPU Threaded | 42.24s ± 1.49s | 0.9x | 3 |
| CPU Multiprocess | 19.17s ± 0.35s | 2.0x | 3 |
| Numba CPU | 33.08s ± 0.33s | 1.2x | 3 |
| Numba CUDA | 2.44s ± 0.05s | 15.9x | 3 |
| CuPy GPU | 1.54s ± 0.05s | 25.2x | 3 |



Bar

**S-Bend (L)** (765,441 nodes)

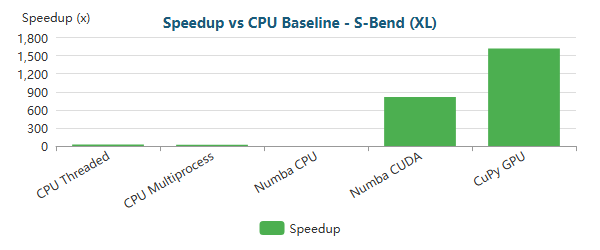
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 50.0s ± 5.0s | 1.0x | 3 |
| CPU Threaded | 3m 23.8s ± 1.6s | 2.0x | 3 |
| CPU Multiprocess | 3m 9.0s ± 8.6s | 2.2x | 3 |
| Numba CPU | 4m 53.3s ± 5.4s | 1.4x | 3 |
| Numba CUDA | 8.50s ± 0.02s | 48.2x | 3 |
| CuPy GPU | 4.19s ± 0.00s | 97.8x | 3 |



Bar

**S-Bend (XL)** (1,286,039 nodes)

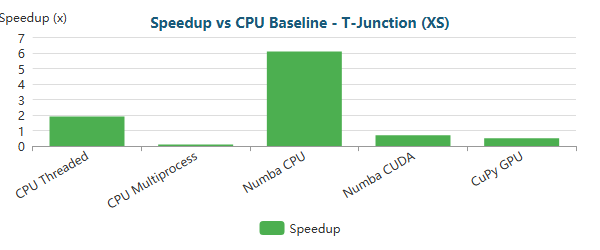
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 209m 12.9s ± 680.6s | 1.0x | 3 |
| CPU Threaded | 7m 55.3s ± 2.9s | 26.4x | 3 |
| CPU Multiprocess | 8m 38.2s ± 6.5s | 24.2x | 3 |
| Numba CPU | 190m 31.7s ± 594.3s | 1.1x | 3 |
| Numba CUDA | 15.41s ± 0.14s | 814.8x | 3 |
| CuPy GPU | 7.76s ± 0.02s | 1617.6x | 3 |



Bar

**T-Junction (XS)** (393 nodes)

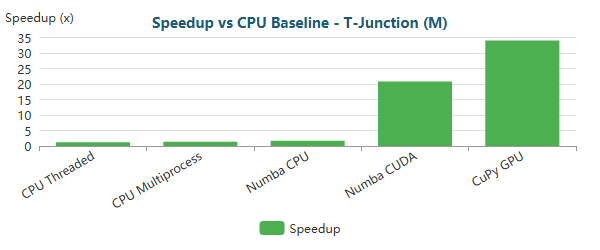
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 35ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 18ms ± 2ms | 1.9x | 3 |
| CPU Multiprocess | 331ms ± 10ms | 0.1x | 3 |
| Numba CPU | <0.01s ± 1ms | 6.1x | 3 |
| Numba CUDA | 48ms ± 4ms | 0.7x | 3 |
| CuPy GPU | 64ms ± 8ms | 0.5x | 3 |



Bar

**T-Junction (M)** (196,420 nodes)

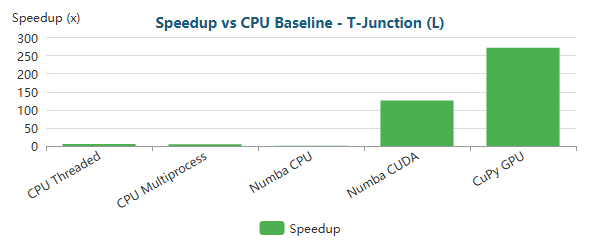
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 48.55s ± 0.08s | 1.0x | 3 |
| CPU Threaded | 41.12s ± 0.61s | 1.2x | 3 |
| CPU Multiprocess | 33.66s ± 1.59s | 1.4x | 3 |
| Numba CPU | 28.00s ± 1.65s | 1.7x | 3 |
| Numba CUDA | 2.33s ± 0.03s | 20.8x | 3 |
| CuPy GPU | 1.43s ± 0.01s | 34.0x | 3 |



Bar

**T-Junction (L)** (768,898 nodes)

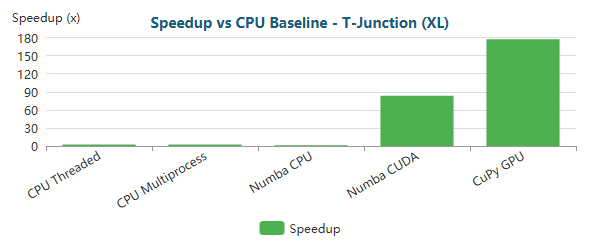
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 17m 33.5s ± 2.5s | 1.0x | 3 |
| CPU Threaded | 3m 9.1s ± 0.3s | 5.6x | 3 |
| CPU Multiprocess | 3m 30.4s ± 2.0s | 5.0x | 3 |
| Numba CPU | 15m 46.9s ± 3.8s | 1.1x | 3 |
| Numba CUDA | 8.37s ± 0.14s | 125.9x | 3 |
| CuPy GPU | 3.88s ± 0.01s | 271.6x | 3 |



Bar

**T-Junction (XL)** (1,291,289 nodes)

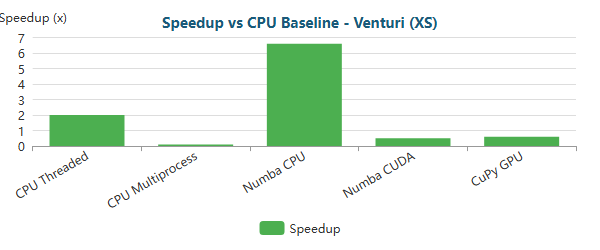
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 20m 17.0s ± 10.9s | 1.0x | 3 |
| CPU Threaded | 7m 19.6s ± 4.4s | 2.8x | 3 |
| CPU Multiprocess | 7m 35.5s ± 24.1s | 2.7x | 3 |
| Numba CPU | 13m 32.2s ± 8.0s | 1.5x | 3 |
| Numba CUDA | 14.56s ± 0.16s | 83.6x | 3 |
| CuPy GPU | 6.87s ± 0.02s | 177.2x | 3 |



Bar

**Venturi (XS)** (341 nodes)

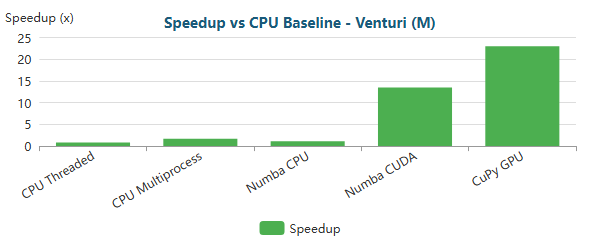
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 30ms ± 2ms | 1.0x | 3 |
| CPU Threaded | 15ms ± 3ms | 2.0x | 3 |
| CPU Multiprocess | 260ms ± 21ms | 0.1x | 3 |
| Numba CPU | <0.01s ± 0ms | 6.6x | 3 |
| Numba CUDA | 60ms ± 9ms | 0.5x | 3 |
| CuPy GPU | 53ms ± 2ms | 0.6x | 3 |



Bar

**Venturi (M)** (194,325 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 32.22s ± 0.16s | 1.0x | 3 |
| CPU Threaded | 38.35s ± 0.25s | 0.8x | 3 |
| CPU Multiprocess | 19.45s ± 0.27s | 1.7x | 3 |
| Numba CPU | 29.96s ± 0.27s | 1.1x | 3 |
| Numba CUDA | 2.38s ± 0.08s | 13.5x | 3 |
| CuPy GPU | 1.40s ± 0.05s | 23.0x | 3 |



Bar

**Venturi (L)** (763,707 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 6.4s ± 0.5s | 1.0x | 3 |
| CPU Threaded | 2m 55.9s ± 0.8s | 2.1x | 3 |
| CPU Multiprocess | 2m 49.2s ± 1.7s | 2.2x | 3 |
| Numba CPU | 4m 29.8s ± 4.4s | 1.4x | 3 |
| Numba CUDA | 7.75s ± 0.08s | 47.3x | 3 |
| CuPy GPU | 3.72s ± 0.01s | 98.6x | 3 |



Bar

**Venturi (XL)** (1,284,412 nodes)

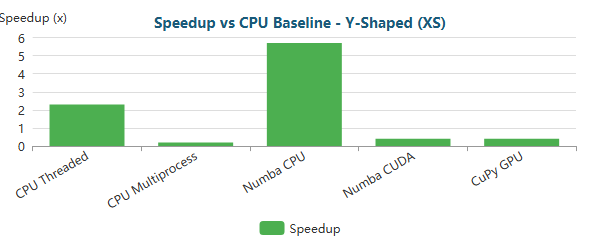
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 16m 51.4s ± 10.0s | 1.0x | 3 |
| CPU Threaded | 6m 39.0s ± 2.9s | 2.5x | 3 |
| CPU Multiprocess | 6m 45.5s ± 2.5s | 2.5x | 3 |
| Numba CPU | 13m 13.4s ± 10.4s | 1.3x | 3 |
| Numba CUDA | 14.48s ± 0.19s | 69.9x | 3 |
| CuPy GPU | 6.51s ± 0.00s | 155.4x | 3 |



Bar

**Y-Shaped (XS)** (201 nodes)

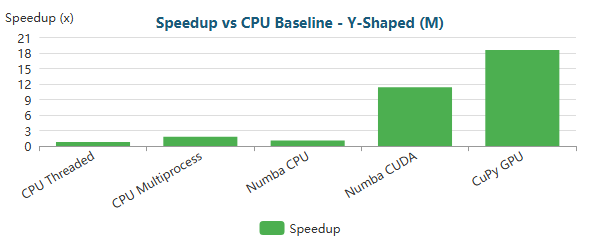
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 22ms ± 2ms | 1.0x | 3 |
| CPU Threaded | <0.01s ± 1ms | 2.3x | 3 |
| CPU Multiprocess | 119ms ± 6ms | 0.2x | 3 |
| Numba CPU | <0.01s ± 0ms | 5.7x | 3 |
| Numba CUDA | 53ms ± 13ms | 0.4x | 3 |
| CuPy GPU | 52ms ± 6ms | 0.4x | 3 |



Bar

**Y-Shaped (M)** (195,853 nodes)

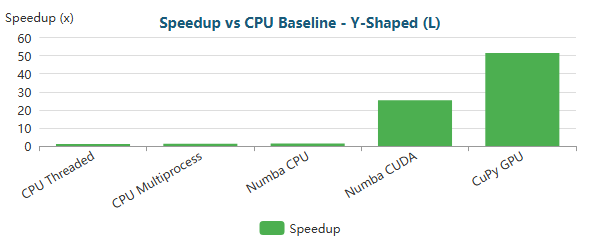
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 25.05s ± 0.49s | 1.0x | 3 |
| CPU Threaded | 31.45s ± 0.54s | 0.8x | 3 |
| CPU Multiprocess | 13.77s ± 0.43s | 1.8x | 3 |
| Numba CPU | 22.86s ± 0.32s | 1.1x | 3 |
| Numba CUDA | 2.20s ± 0.08s | 11.4x | 3 |
| CuPy GPU | 1.35s ± 0.01s | 18.6x | 3 |



Bar

**Y-Shaped (L)** (772,069 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 2m 50.4s ± 0.4s | 1.0x | 3 |
| CPU Threaded | 2m 34.5s ± 0.6s | 1.1x | 3 |
| CPU Multiprocess | 2m 9.9s ± 1.2s | 1.3x | 3 |
| Numba CPU | 1m 57.7s ± 0.1s | 1.4x | 3 |
| Numba CUDA | 6.73s ± 0.13s | 25.3x | 3 |
| CuPy GPU | 3.31s ± 0.00s | 51.4x | 3 |



Bar

**Y-Shaped (XL)** (1,357,953 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 53.4s ± 10.2s | 1.0x | 3 |
| CPU Threaded | 6m 28.4s ± 1.3s | 1.1x | 3 |
| CPU Multiprocess | 5m 54.8s ± 2.0s | 1.2x | 3 |
| Numba CPU | 5m 24.6s ± 4.1s | 1.3x | 3 |
| Numba CUDA | 12.85s ± 0.23s | 32.2x | 3 |
| CuPy GPU | 6.09s ± 0.01s | 67.9x | 3 |



Bar

### 1.2.1 Critical Analysis RTX 4090

#### 1.2.1.1 Small-Scale Problems (XS meshes)

Across all geometries with **XS meshes (≈200–400 nodes)**, the benchmarks show a **clear overhead-dominated regime**:

* **CPU-based implementations dominate** absolute performance.
* **Numba JIT CPU** is consistently the fastest approach, delivering speedups of approximately **~5.5× to ~6.6×** versus the CPU baseline (e.g., **6.1×** for Backward-Facing Step, **5.5×** for Elbow 90°, **5.8×** for S-Bend, **6.1×** for T-Junction, **6.6×** for Venturi, **5.7×** for Y-shaped).
* **GPU implementations (Numba CUDA and CuPy GPU)** are systematically *slower* than the CPU baseline, with speedups typically in the **~0.4× to ~0.9×** range (real slowdowns).

This behavior is technically expected. At XS scale, FEM execution time is dominated by fixed GPU overheads—kernel launch latency, GPU context/synchronization, and any host–device management costs—which cannot be amortized with only a few hundred nodes. Even when GPU kernels execute quickly, the end-to-end runtime remains constrained by these constant terms, confirming that **GPU acceleration is not advantageous for small FEM problems**.

Additionally, **CPU multiprocessing performs poorly** in this regime, with speedups around **~0.1×–0.2×** (e.g., 260–374 ms and similar), because process spawning, IPC, and serialization overhead dominate the computation. This reinforces that parallel execution models must be matched to the problem scale.

#### 1.2.1.2 Medium-Scale Problems (M meshes)

For **medium-scale meshes (≈160k–200k nodes)**, the performance profile changes sharply and consistently across geometries, indicating a clear **CPU–GPU crossover**:

* **GPU acceleration becomes dominant** and stable across all cases.
* **Numba CUDA** reaches speedups of approximately **~11× to ~22×** (e.g., **11.4×** Y-shaped, **13.5×** Venturi, **15.9×** S-bend, **19.4×** backward-facing step, **20.8×** T-junction, **22.1×** elbow).
* **CuPy GPU (RawKernel)** consistently outperforms Numba CUDA, delivering speedups around **~18× to ~42×** (e.g., **18.6×** Y-shaped, **23.0×** Venturi, **25.2×** S-bend, **34.0×** T-junction, **36.6×** backward-facing step, **41.7×** elbow).
* **CPU-based approaches saturate**, typically remaining around **~1.1× to ~1.8×** (and sometimes below 1× for threading in specific cases), even with multiprocessing or JIT.

At this scale, the workload becomes large enough to exploit GPU parallelism effectively: assembly and post-processing costs are substantially reduced and the overall runtime becomes increasingly solver-dominated. Geometry-dependent differences persist, with the strongest GPU gains observed in cases that increase solver workload and sparsity complexity, consistent with improved GPU utilization as arithmetic intensity rises.

#### 1.2.1.3 Large-Scale and Extreme-Scale Problems (L and XL meshes)

With **large (L) and extra-large (XL) meshes (≈600k–1.35M nodes)**, GPU acceleration shifts from beneficial to **critical**:

* **CPU baseline runtimes grow to multiple minutes**, making CPU-only execution increasingly impractical for iterative experimentation or production-scale runs.
* **Threading and multiprocessing provide limited and inconsistent relief**, typically around **~1.1×–2.6×**, with some higher outliers driven by an extreme baseline (notably the S-Bend XL case, where the baseline exhibits very large variance).
* **Numba JIT CPU often approaches baseline**, indicating a memory-bandwidth constrained regime where JIT removes Python overhead but cannot overcome sparse-memory access and bandwidth limits.
* **Numba CUDA delivers large speedups**, typically **~25×–126×** for L and **~32×–84×** for XL across the geometries reported.
* **CuPy GPU defines the performance envelope**, reaching approximately **~51×–272×** for L and **~68×–177×** for XL on the reported cases.

At these scales, assembly and post-processing are effectively amortized and cease to be limiting factors. Performance becomes constrained primarily by the **sparse iterative solver**, which is fundamentally **memory-bandwidth bound** and sensitive to irregular sparsity patterns. The persistence of a solver-dominated runtime explains why gains do not scale linearly with problem size or purely with compute throughput.

#### 1.2.1.4 Comparative Assessment Across Scales

From a practical standpoint, the benchmarks support the following execution-model selection:

* **XS meshes:** Numba JIT CPU — minimal overhead and compiled execution.
* **M meshes:** CuPy GPU (RawKernel) — best throughput and consistent GPU advantage.
* **L and XL meshes:** CuPy GPU (RawKernel) — maximum scalability and best end-to-end runtime.
* **CPU-only environments:** Numba JIT CPU — best balance of speed and portability.
* **GPU prototyping:** Numba CUDA — easier development with strong speedups.
* **Production GPU workloads:** CuPy RawKernel — highest and most consistent speedups.

#### 1.2.1.5 Comparative Assessment of Execution Models

From a practical standpoint, the benchmark results support the following conclusions:

| Regime | Best Execution Model | Rationale |
| --- | --- | --- |
| XS meshes | Numba JIT CPU | Minimal overhead, compiled execution |
| M meshes | CuPy GPU (RawKernel) | Maximum throughput, full GPU residency |
| CPU-only environments | Numba JIT CPU | Best balance of speed and memory efficiency |
| GPU prototyping | Numba CUDA | Easier development, acceptable performance |
| Production GPU | CuPy RawKernel | Highest and most consistent speedups |

The RTX 4090 demonstrates **strong scalability** once the problem size justifies GPU usage. However, the results reinforce a core insight: **hardware capability alone is insufficient**—algorithmic structure, solver behavior, memory access patterns, and execution model determine the realized speedups.

Overall, the benchmarks confirm that the RTX 4090 is highly effective for **medium to extreme-scale FEM simulations**, delivering order-of-magnitude speedups when the workload is large enough to amortize overheads. At the same time, the data reinforces several critical best practices:

* GPU acceleration should be **selectively applied**, rather than used indiscriminately.
* Small and interactive FEM problems are better served by optimized CPU execution.
* For large-scale production workloads, **RawKernel-based GPU implementations provide the highest return on investment**.
* At scale, the **sparse linear solver—not the assembly kernel—becomes the dominant bottleneck**, and further gains require solver-level improvements (e.g., better preconditioning, fewer iterations, or alternative sparse methods).

Tthe RTX 4090 as a robust GPU for FEM acceleration across meaningful problem sizes, while confirming that the ultimate ceiling is governed by sparse linear algebra efficiency rather than raw compute throughput.

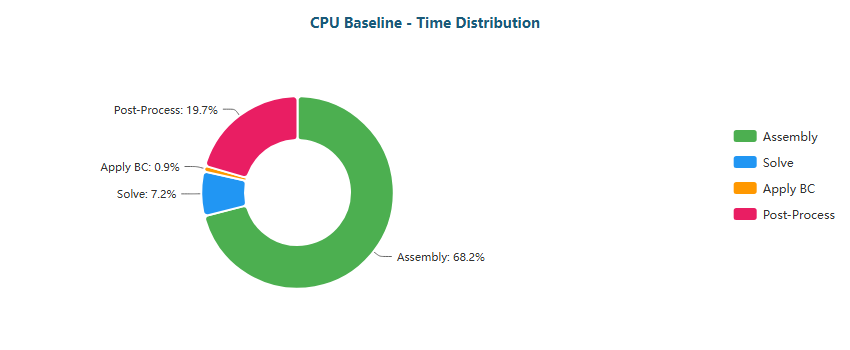
### 1.2.2 RTX 4090 Bottleneck Evolution Critical Analysis

As optimizations progress, the computational bottleneck shifts:

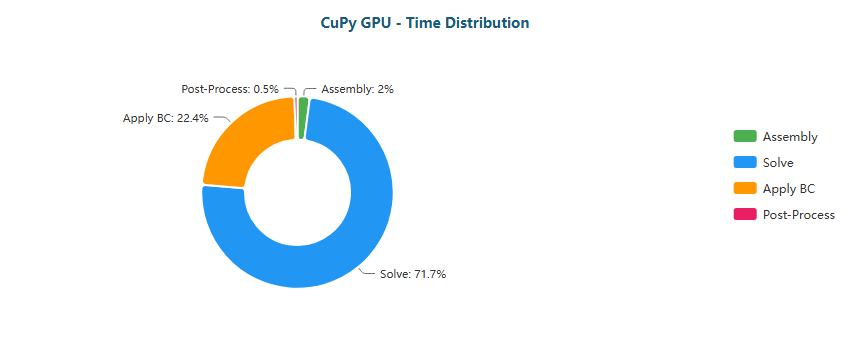
#### Backward-Facing Step (XS) - 287 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (68%) | Post-Proc (20%) |
| CPU Threaded | Assembly (52%) | Post-Proc (21%) |
| CPU Multiprocess | Assembly (52%) | Post-Proc (46%) |
| Numba CPU | Solve (45%) | Assembly (14%) |
| Numba CUDA | Solve (87%) | Assembly (6%) |
| CuPy GPU | Solve (72%) | BC (22%) |

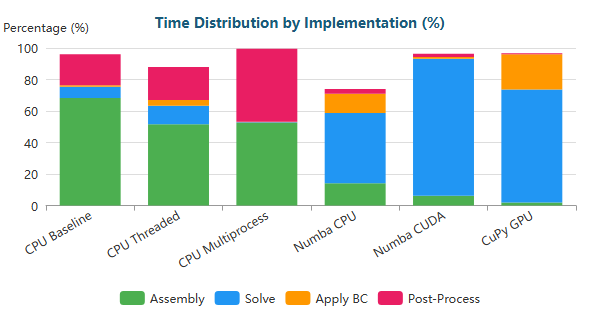
**Time Distribution:**



Pie



Pie

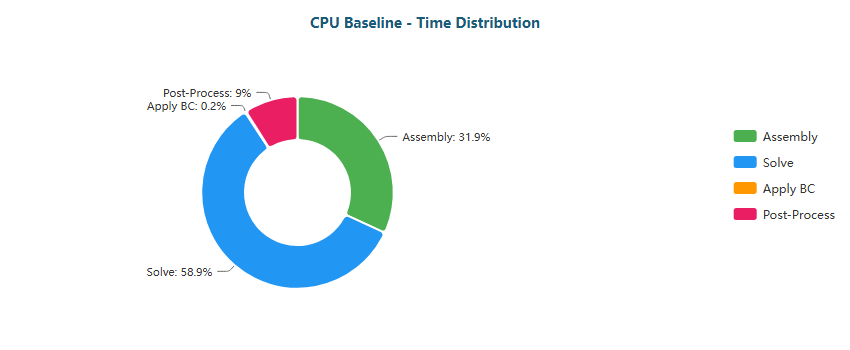


Stacked Bar

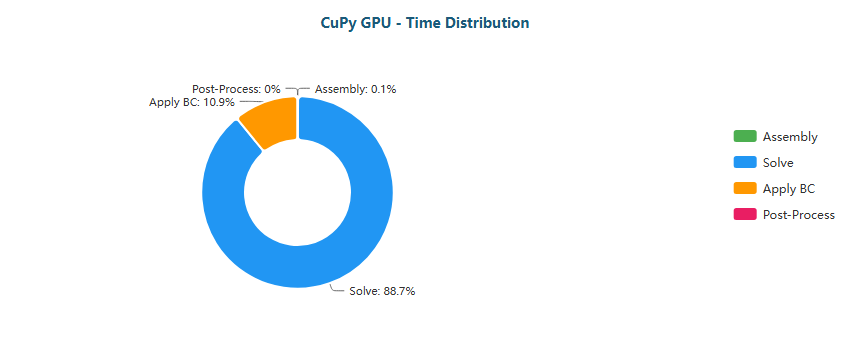
#### Backward-Facing Step (M) - 195,362 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (59%) | Assembly (32%) |
| CPU Threaded | Solve (67%) | Assembly (19%) |
| CPU Multiprocess | Solve (81%) | BC (16%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (49%) | Assembly (30%) |
| CuPy GPU | Solve (89%) | BC (11%) |

**Time Distribution:**



Pie



Pie

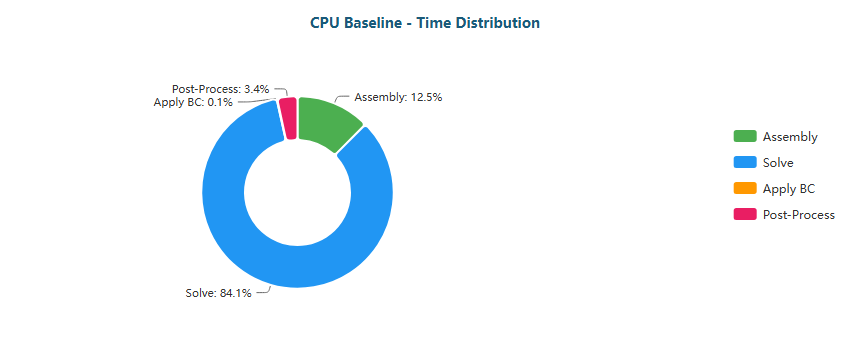


Stacked Bar

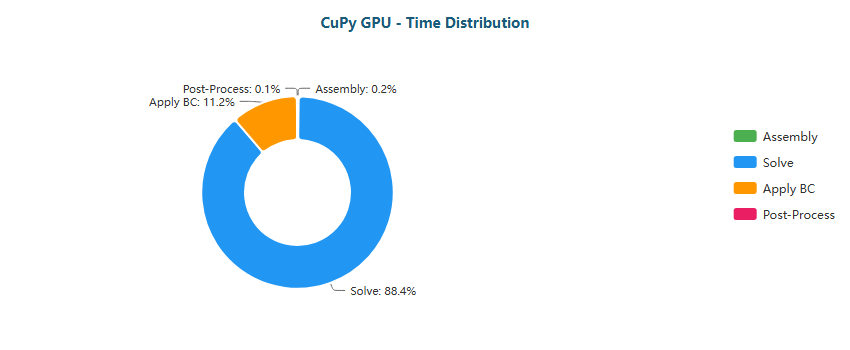
#### Backward-Facing Step (L) - 766,088 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (84%) | Assembly (12%) |
| CPU Threaded | Solve (73%) | Assembly (16%) |
| CPU Multiprocess | Solve (83%) | BC (16%) |
| Numba CPU | Solve (99%) | BC (0%) |
| Numba CUDA | Solve (43%) | Assembly (34%) |
| CuPy GPU | Solve (88%) | BC (11%) |

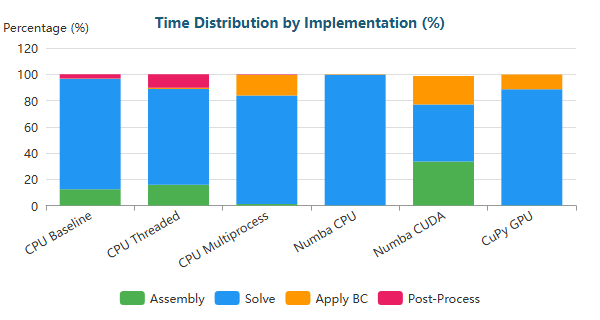
**Time Distribution:**



Pie



Pie

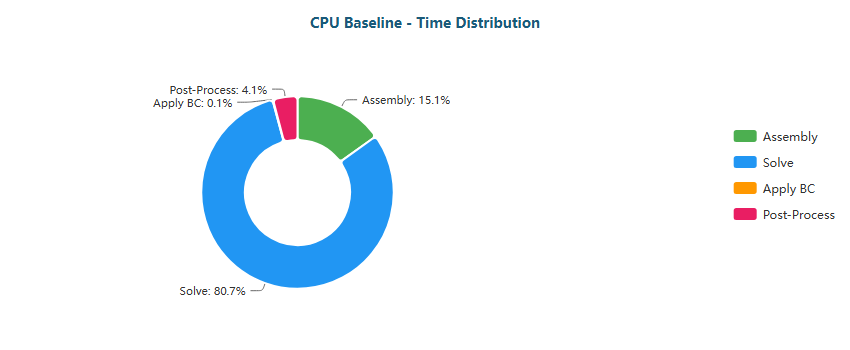


Stacked Bar

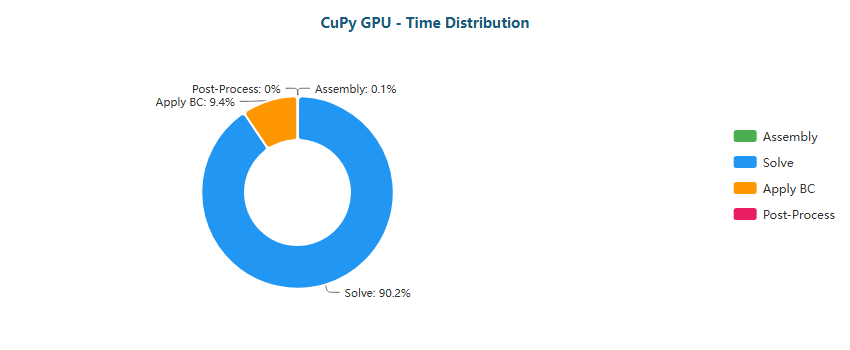
#### Backward-Facing Step (XL) - 1,283,215 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (81%) | Assembly (15%) |
| CPU Threaded | Solve (80%) | Assembly (12%) |
| CPU Multiprocess | Solve (84%) | BC (14%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (44%) | Assembly (33%) |
| CuPy GPU | Solve (90%) | BC (9%) |

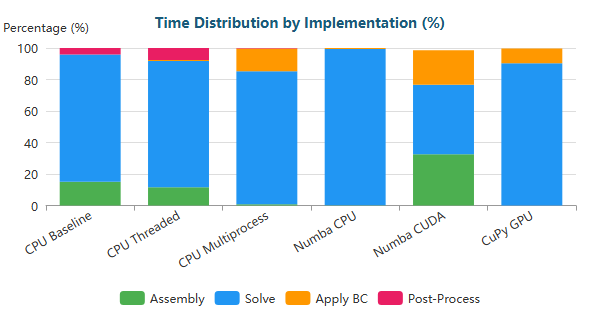
**Time Distribution:**



Pie



Pie

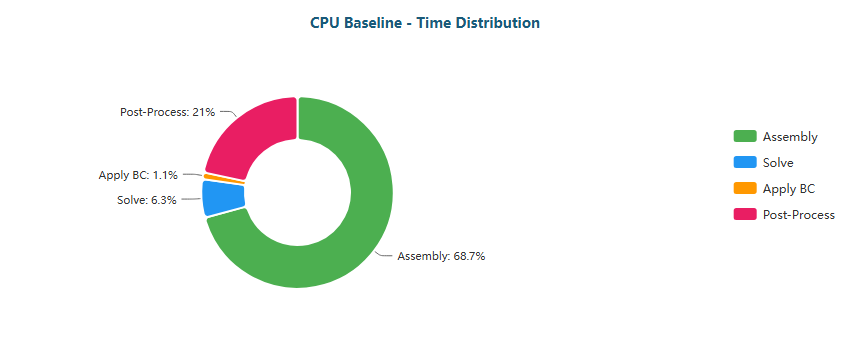


Stacked Bar

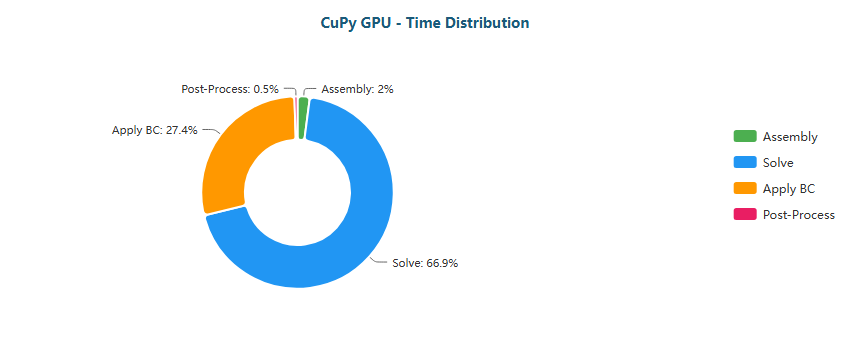
#### Elbow 90° (XS) - 411 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (69%) | Post-Proc (21%) |
| CPU Threaded | Assembly (53%) | Post-Proc (24%) |
| CPU Multiprocess | Assembly (53%) | Post-Proc (45%) |
| Numba CPU | Solve (39%) | BC (16%) |
| Numba CUDA | Solve (86%) | Assembly (7%) |
| CuPy GPU | Solve (67%) | BC (27%) |

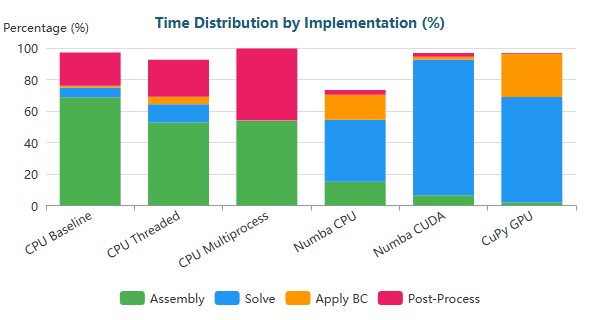
**Time Distribution:**



Pie



Pie

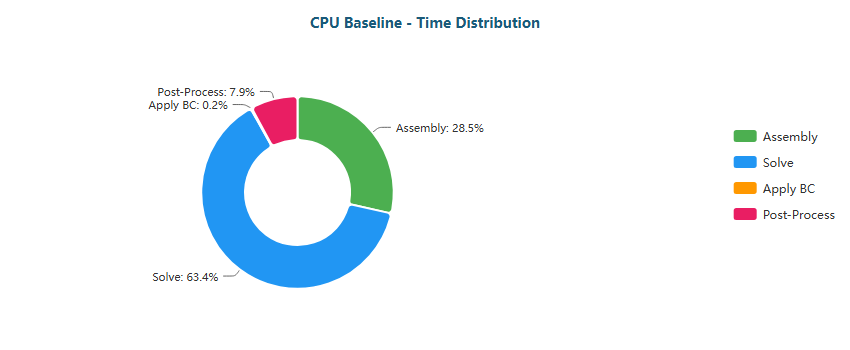


Stacked Bar

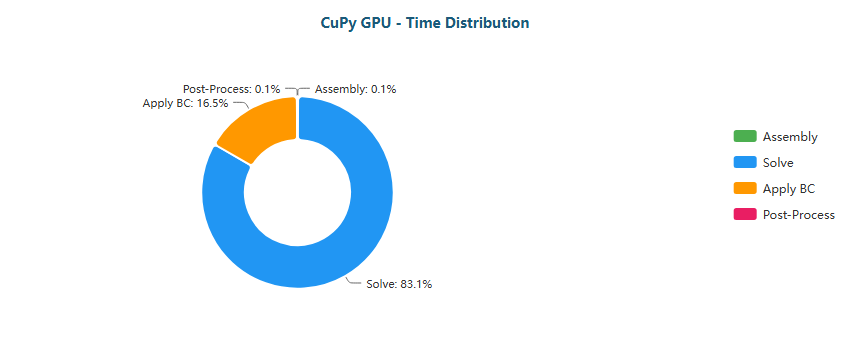
#### Elbow 90° (M) - 161,984 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (63%) | Assembly (28%) |
| CPU Threaded | Solve (64%) | Assembly (21%) |
| CPU Multiprocess | Solve (65%) | BC (32%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (49%) | Assembly (31%) |
| CuPy GPU | Solve (83%) | BC (16%) |

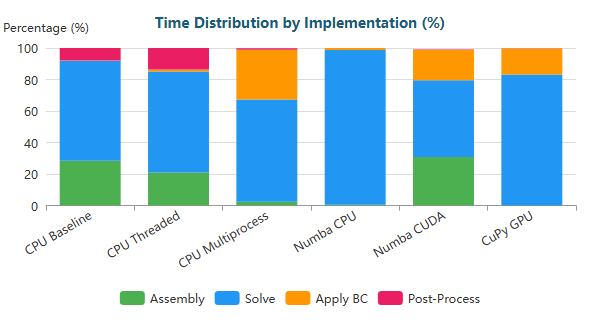
**Time Distribution:**



Pie



Pie

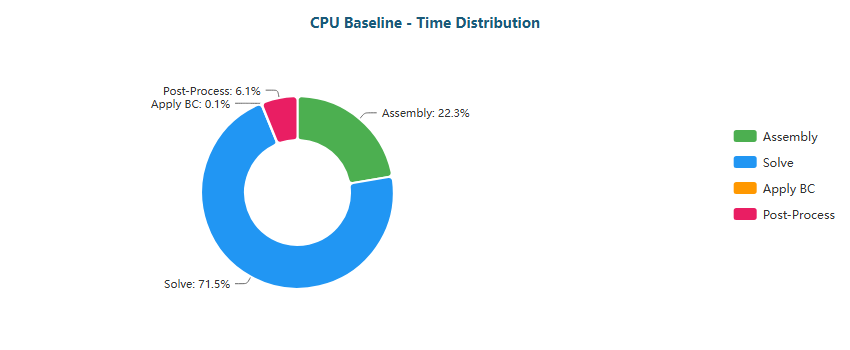


Stacked Bar

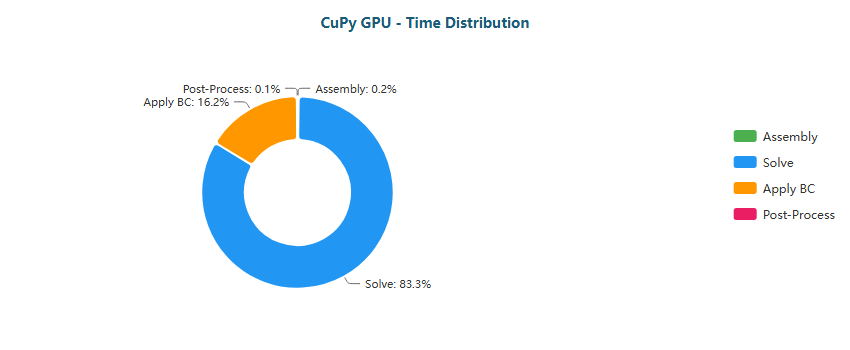
#### Elbow 90° (L) - 623,153 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (71%) | Assembly (22%) |
| CPU Threaded | Solve (66%) | Assembly (19%) |
| CPU Multiprocess | Solve (59%) | BC (39%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (40%) | Assembly (37%) |
| CuPy GPU | Solve (83%) | BC (16%) |

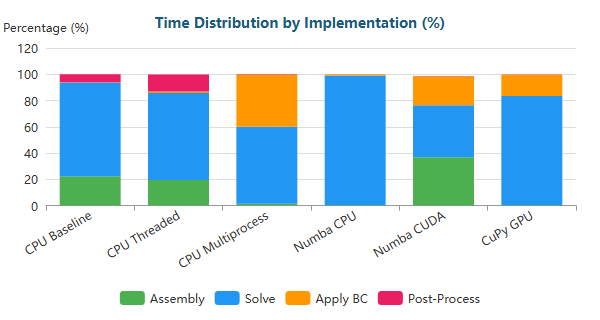
**Time Distribution:**



Pie



Pie

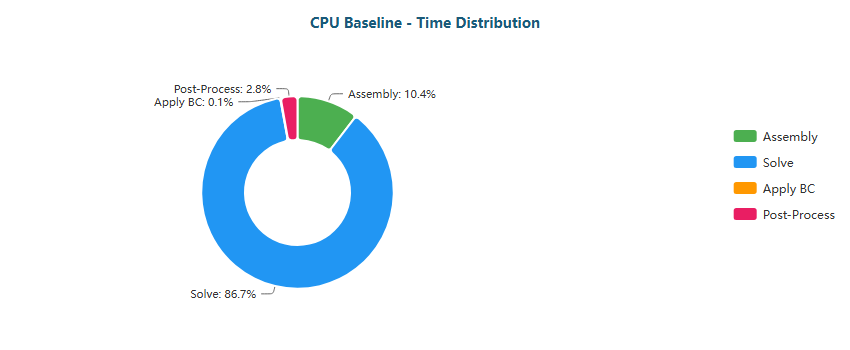


Stacked Bar

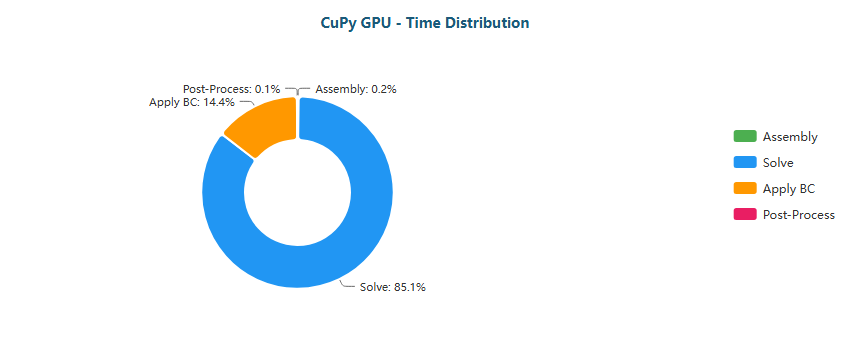
#### Elbow 90° (XL) - 1,044,857 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (87%) | Assembly (10%) |
| CPU Threaded | Solve (75%) | Assembly (15%) |
| CPU Multiprocess | Solve (59%) | BC (40%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (40%) | Assembly (36%) |
| CuPy GPU | Solve (85%) | BC (14%) |

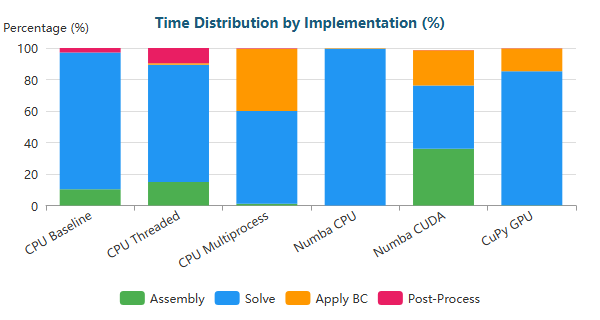
**Time Distribution:**



Pie



Pie

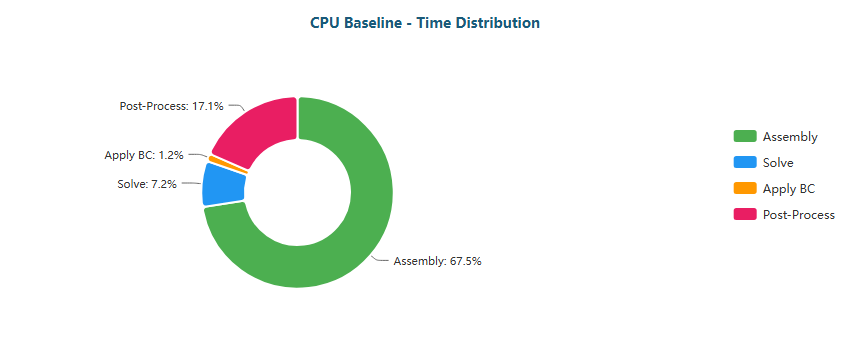


Stacked Bar

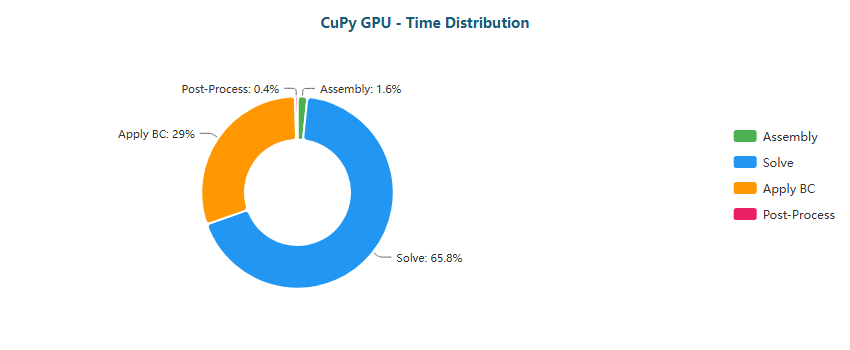
#### S-Bend (XS) - 387 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (68%) | Post-Proc (17%) |
| CPU Threaded | Assembly (55%) | Post-Proc (17%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (48%) |
| Numba CPU | Solve (39%) | BC (16%) |
| Numba CUDA | Solve (91%) | Assembly (4%) |
| CuPy GPU | Solve (66%) | BC (29%) |

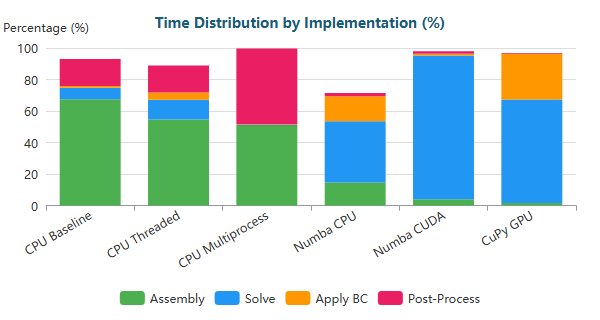
**Time Distribution:**



Pie



Pie

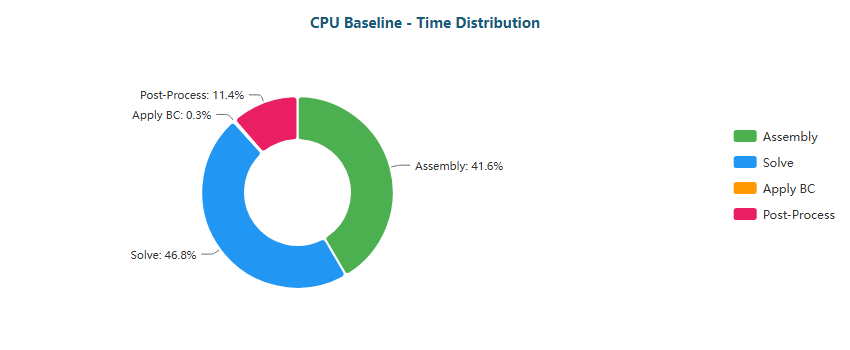


Stacked Bar

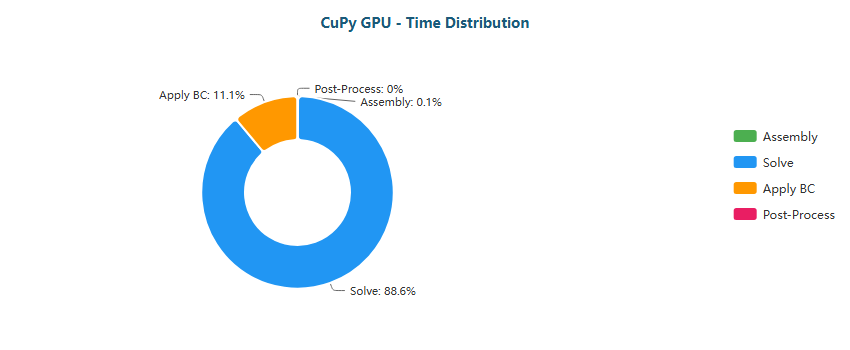
#### S-Bend (M) - 196,078 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (47%) | Assembly (42%) |
| CPU Threaded | Solve (71%) | Assembly (17%) |
| CPU Multiprocess | Solve (77%) | BC (18%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (53%) | Assembly (28%) |
| CuPy GPU | Solve (89%) | BC (11%) |

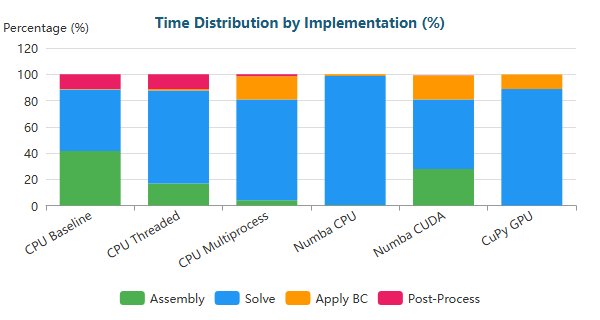
**Time Distribution:**



Pie



Pie

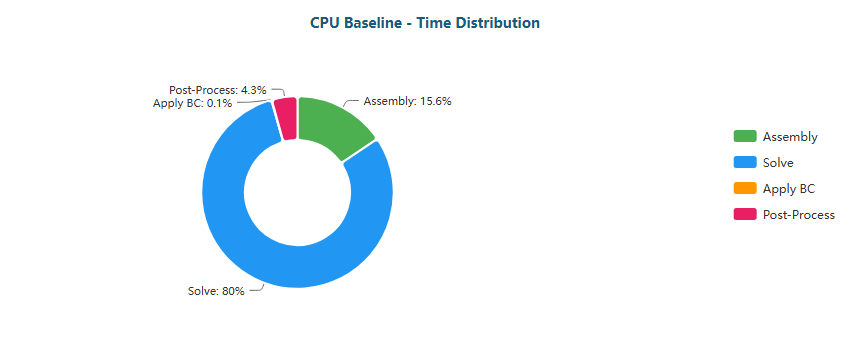


Stacked Bar

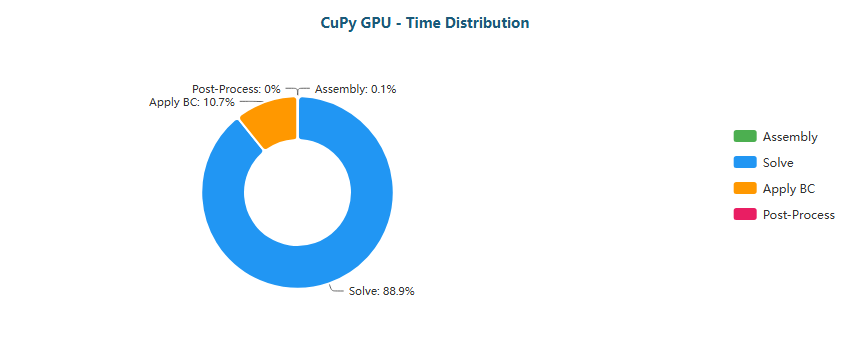
#### S-Bend (L) - 765,441 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (80%) | Assembly (16%) |
| CPU Threaded | Solve (76%) | Assembly (14%) |
| CPU Multiprocess | Solve (85%) | BC (13%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (46%) | Assembly (32%) |
| CuPy GPU | Solve (89%) | BC (11%) |

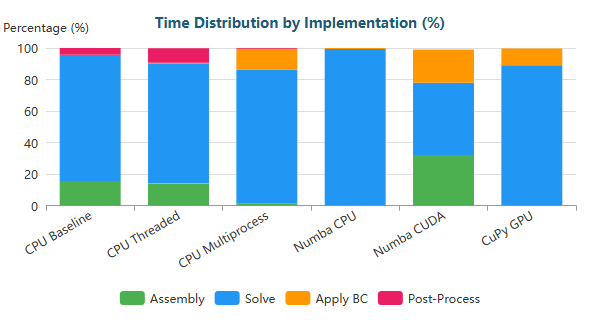
**Time Distribution:**



Pie



Pie

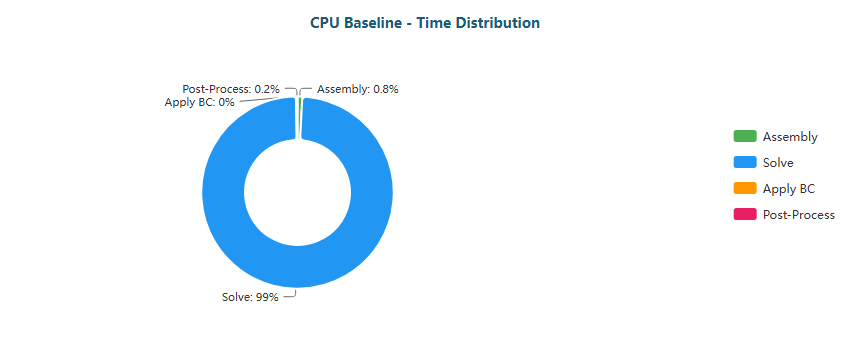


Stacked Bar

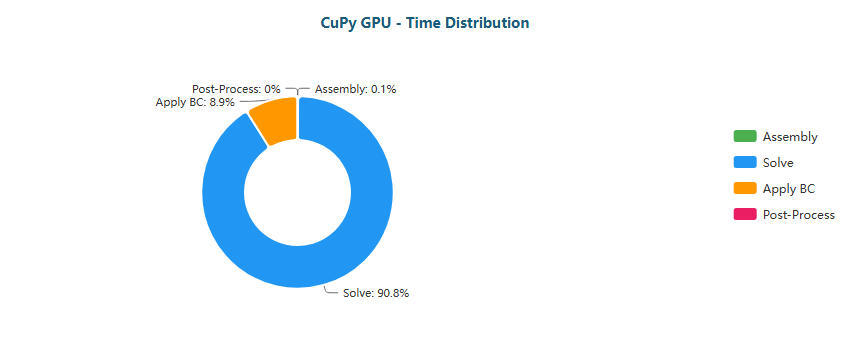
#### S-Bend (XL) - 1,286,039 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (99%) | Assembly (1%) |
| CPU Threaded | Solve (82%) | Assembly (10%) |
| CPU Multiprocess | Solve (89%) | BC (9%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (49%) | Assembly (30%) |
| CuPy GPU | Solve (91%) | BC (9%) |

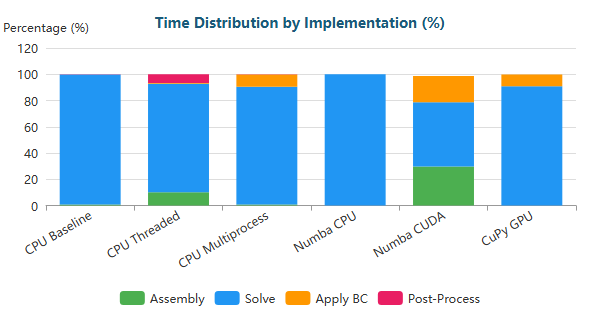
**Time Distribution:**



Pie



Pie

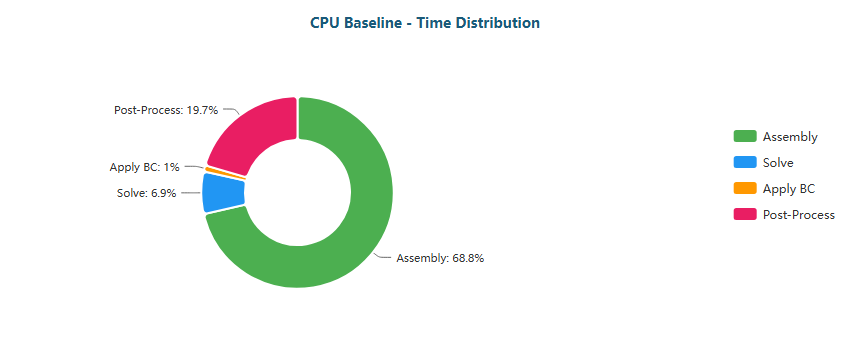


Stacked Bar

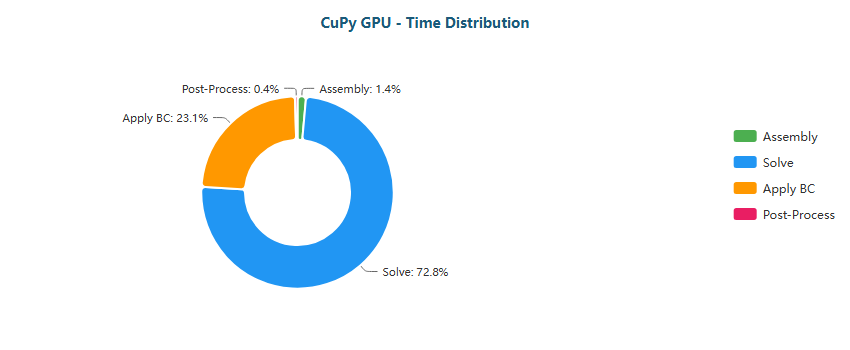
#### T-Junction (XS) - 393 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (69%) | Post-Proc (20%) |
| CPU Threaded | Assembly (52%) | Post-Proc (23%) |
| CPU Multiprocess | Assembly (53%) | Post-Proc (46%) |
| Numba CPU | Solve (41%) | Assembly (15%) |
| Numba CUDA | Solve (85%) | Assembly (8%) |
| CuPy GPU | Solve (73%) | BC (23%) |

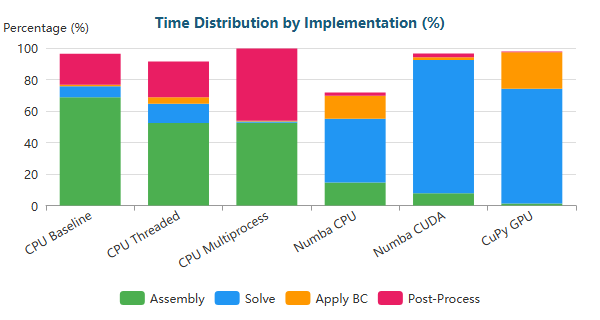
**Time Distribution:**



Pie



Pie



Stacked Bar

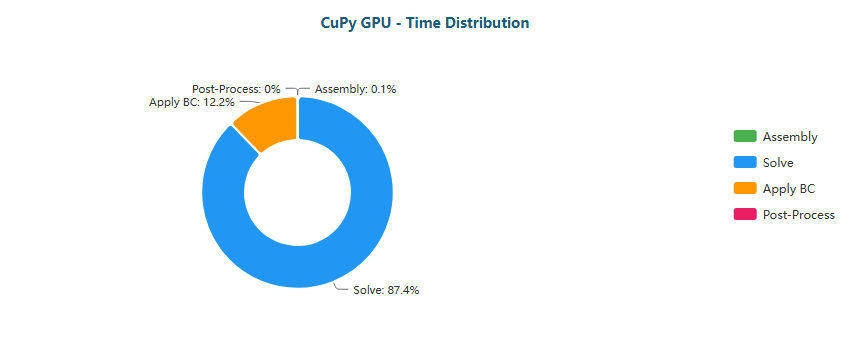
#### T-Junction (M) - 196,420 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (60%) | Assembly (31%) |
| CPU Threaded | Solve (70%) | Assembly (18%) |
| CPU Multiprocess | Solve (86%) | BC (11%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (51%) | Assembly (29%) |
| CuPy GPU | Solve (87%) | BC (12%) |

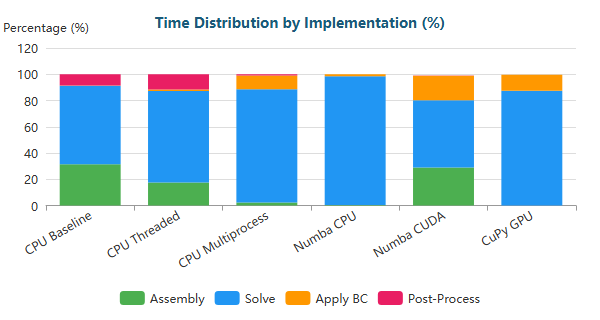
**Time Distribution:**



Pie



Pie



Stacked Bar

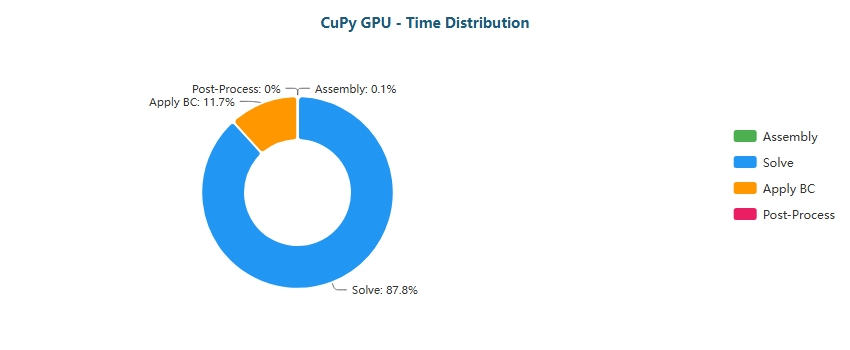
#### T-Junction (L) - 768,898 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (93%) | Assembly (5%) |
| CPU Threaded | Solve (74%) | Assembly (15%) |
| CPU Multiprocess | Solve (87%) | BC (11%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (45%) | Assembly (33%) |
| CuPy GPU | Solve (88%) | BC (12%) |

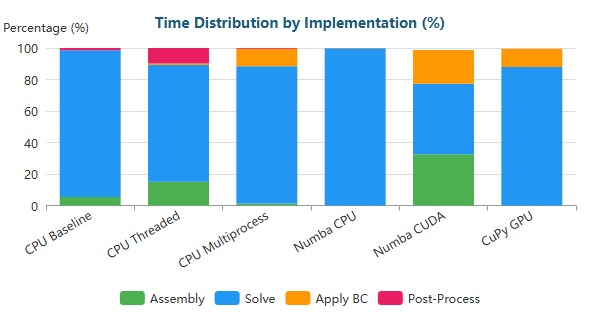
**Time Distribution:**



Pie



Pie

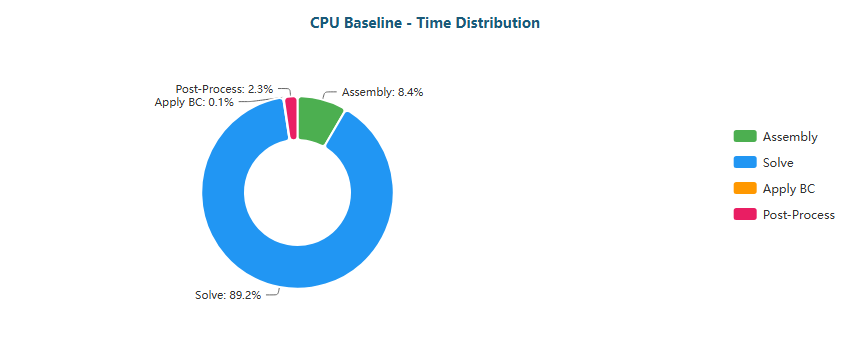


Stacked Bar

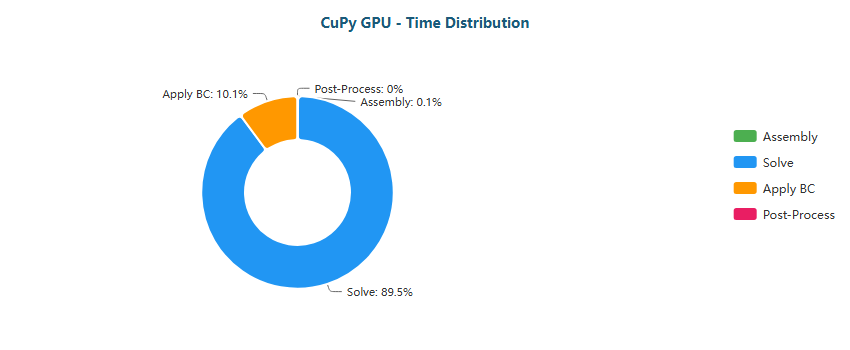
#### T-Junction (XL) - 1,291,289 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (89%) | Assembly (8%) |
| CPU Threaded | Solve (81%) | Assembly (11%) |
| CPU Multiprocess | Solve (88%) | BC (11%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (46%) | Assembly (32%) |
| CuPy GPU | Solve (90%) | BC (10%) |

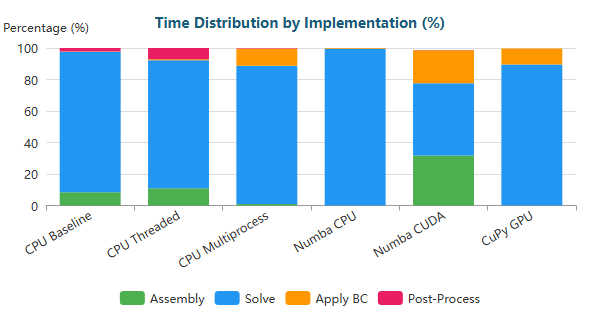
**Time Distribution:**



Pie



Pie

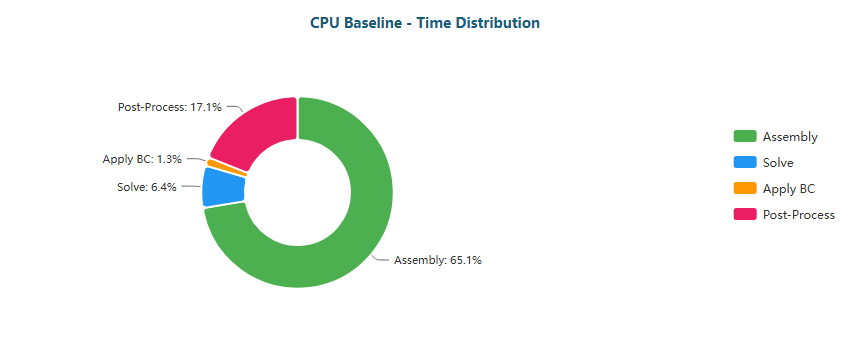


Stacked Bar

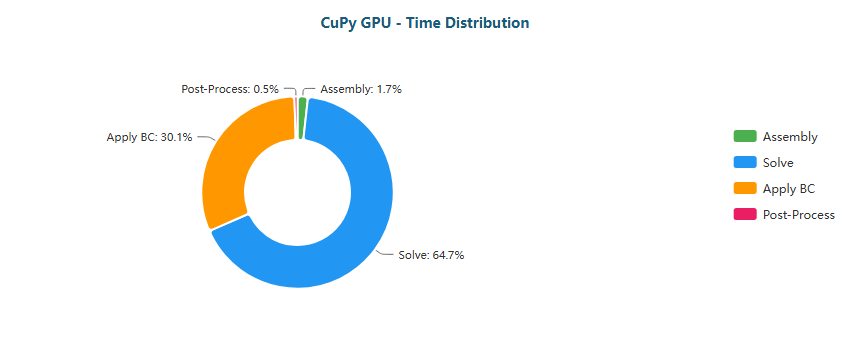
#### Venturi (XS) - 341 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (65%) | Post-Proc (17%) |
| CPU Threaded | Assembly (50%) | Post-Proc (23%) |
| CPU Multiprocess | Assembly (51%) | Post-Proc (47%) |
| Numba CPU | Solve (43%) | BC (16%) |
| Numba CUDA | Solve (90%) | Assembly (5%) |
| CuPy GPU | Solve (65%) | BC (30%) |

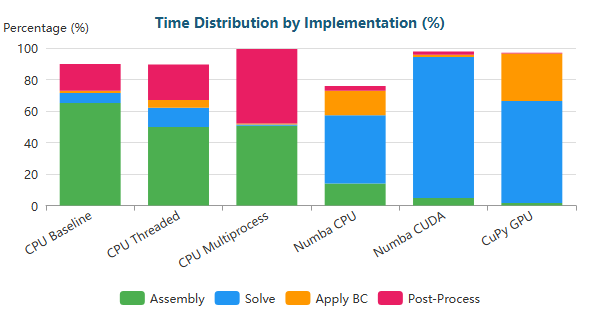
**Time Distribution:**



Pie



Pie

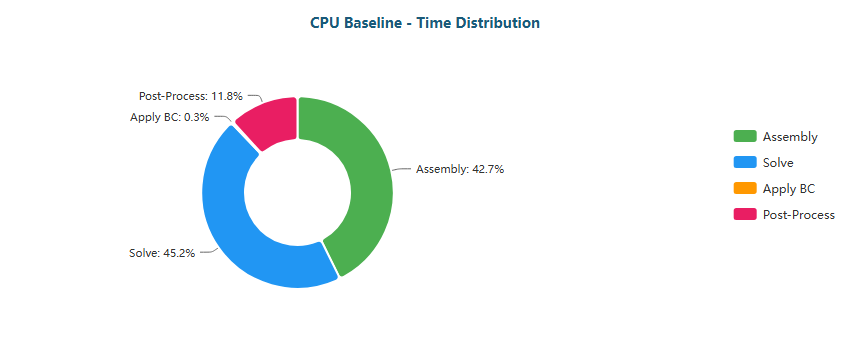


Stacked Bar

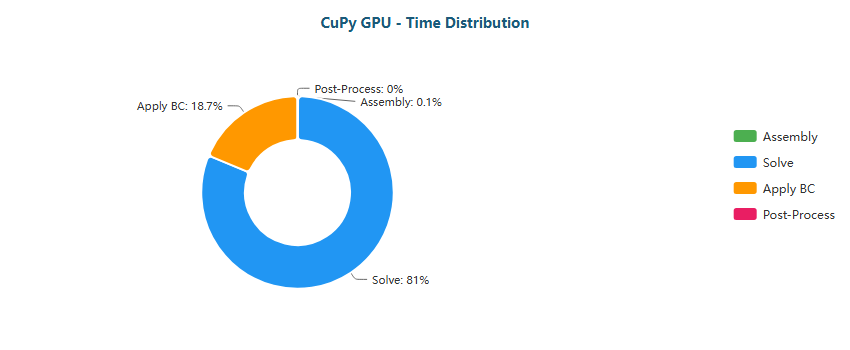
#### Venturi (M) - 194,325 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (45%) | Assembly (43%) |
| CPU Threaded | Solve (68%) | Assembly (19%) |
| CPU Multiprocess | Solve (65%) | BC (30%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (48%) | Assembly (29%) |
| CuPy GPU | Solve (81%) | BC (19%) |

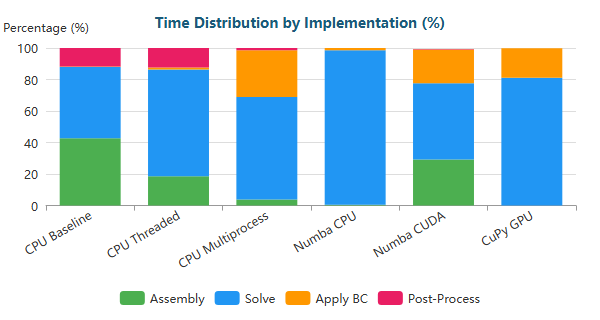
**Time Distribution:**



Pie



Pie



Stacked Bar

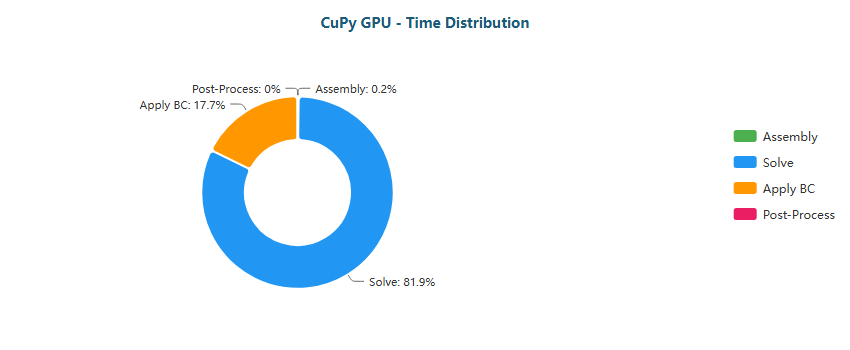
#### Venturi (L) - 763,707 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (78%) | Assembly (17%) |
| CPU Threaded | Solve (72%) | Assembly (16%) |
| CPU Multiprocess | Solve (73%) | BC (25%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (42%) | Assembly (35%) |
| CuPy GPU | Solve (82%) | BC (18%) |

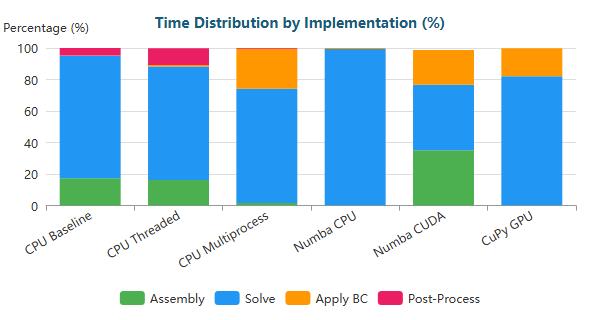
**Time Distribution:**



Pie



Pie

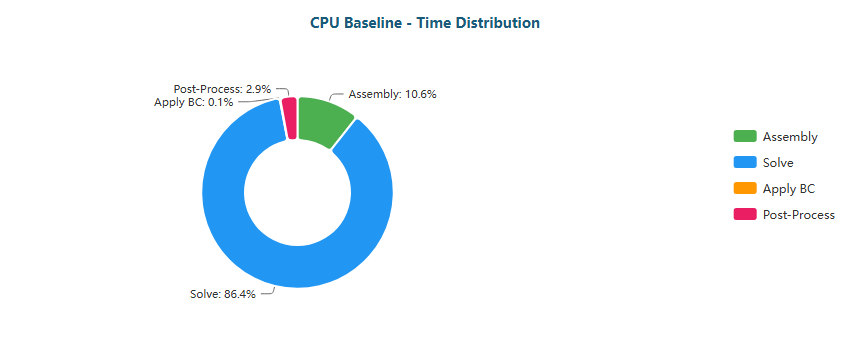


Stacked Bar

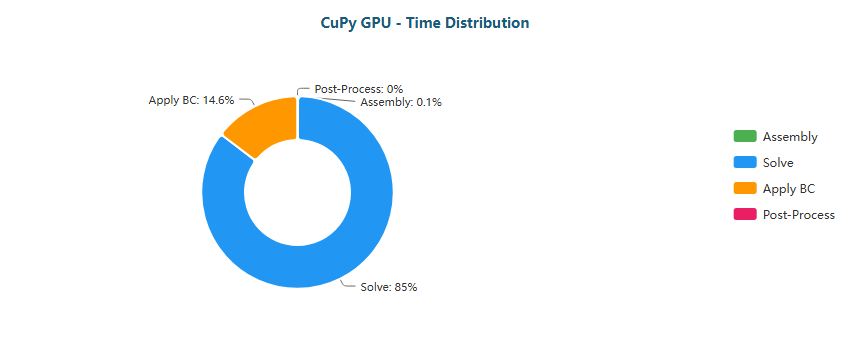
#### Venturi (XL) - 1,284,412 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (86%) | Assembly (11%) |
| CPU Threaded | Solve (79%) | Assembly (12%) |
| CPU Multiprocess | Solve (76%) | BC (22%) |
| Numba CPU | Solve (99%) | BC (0%) |
| Numba CUDA | Solve (43%) | Assembly (32%) |
| CuPy GPU | Solve (85%) | BC (15%) |

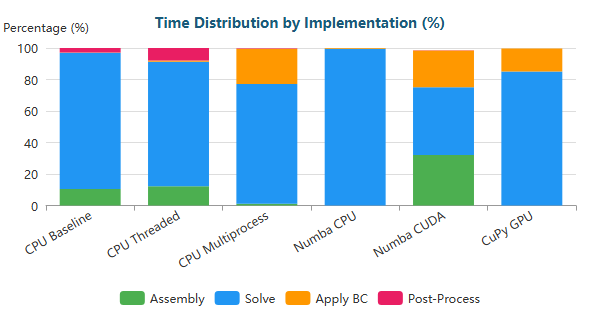
**Time Distribution:**



Pie



Pie

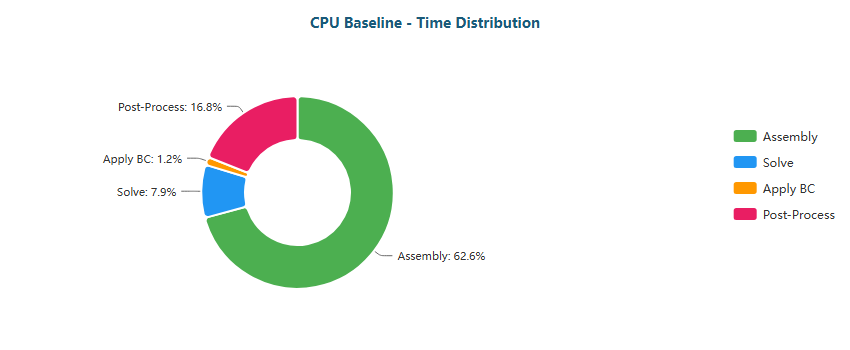


Stacked Bar

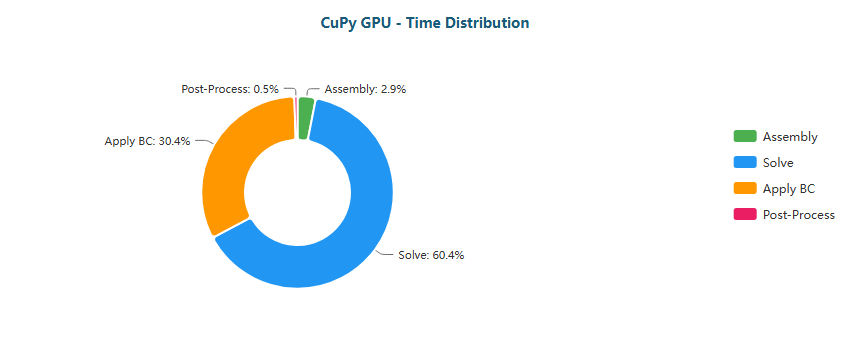
#### Y-Shaped (XS) - 201 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (63%) | Post-Proc (17%) |
| CPU Threaded | Assembly (44%) | Post-Proc (22%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (45%) |
| Numba CPU | Solve (39%) | Assembly (14%) |
| Numba CUDA | Solve (89%) | Assembly (5%) |
| CuPy GPU | Solve (60%) | BC (30%) |

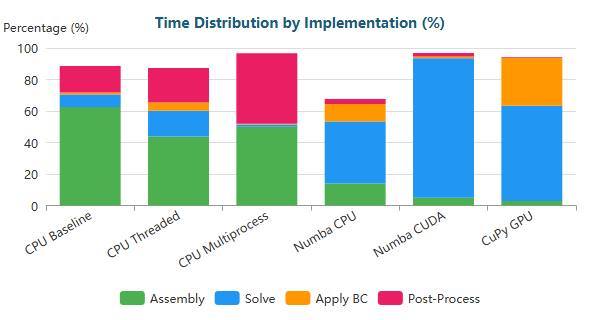
**Time Distribution:**



Pie



Pie

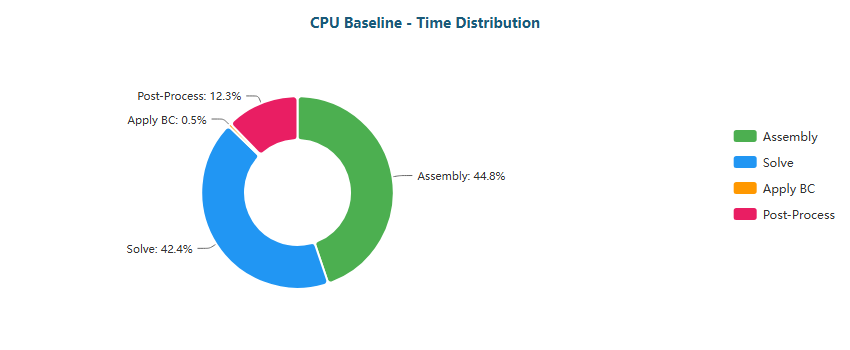


Stacked Bar

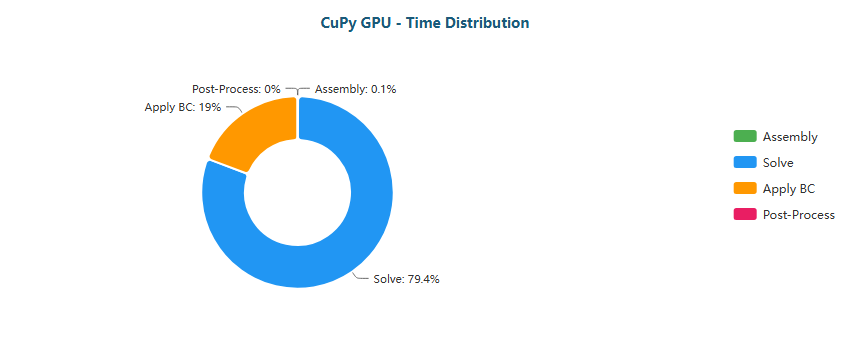
#### Y-Shaped (M) - 195,853 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (45%) | Solve (42%) |
| CPU Threaded | Solve (71%) | Assembly (17%) |
| CPU Multiprocess | Solve (76%) | BC (18%) |
| Numba CPU | Solve (97%) | BC (2%) |
| Numba CUDA | Solve (50%) | Assembly (26%) |
| CuPy GPU | Solve (79%) | BC (19%) |

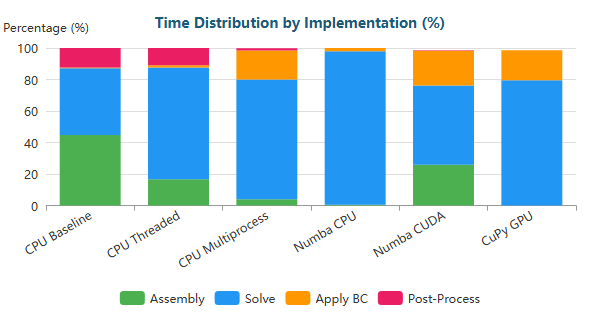
**Time Distribution:**



Pie



Pie

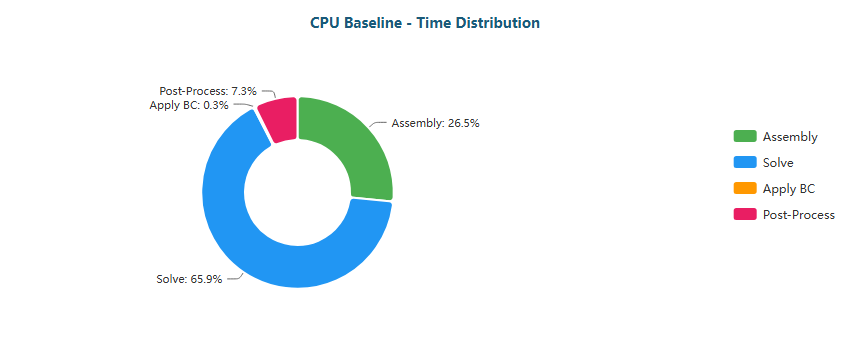


Stacked Bar

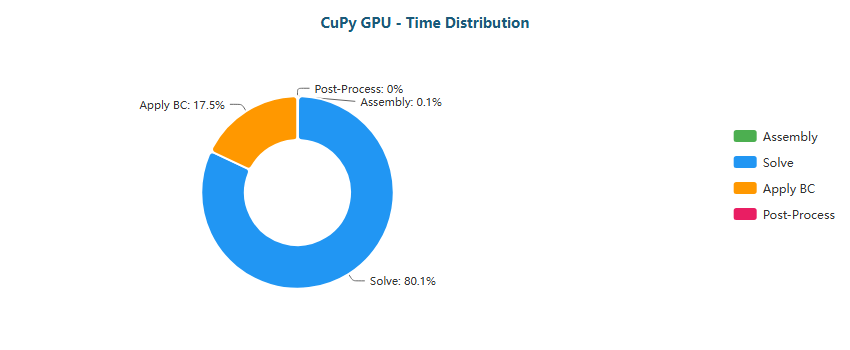
#### Y-Shaped (L) - 772,069 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (66%) | Assembly (26%) |
| CPU Threaded | Solve (75%) | Assembly (14%) |
| CPU Multiprocess | Solve (83%) | BC (15%) |
| Numba CPU | Solve (98%) | BC (1%) |
| Numba CUDA | Solve (41%) | Assembly (32%) |
| CuPy GPU | Solve (80%) | BC (18%) |

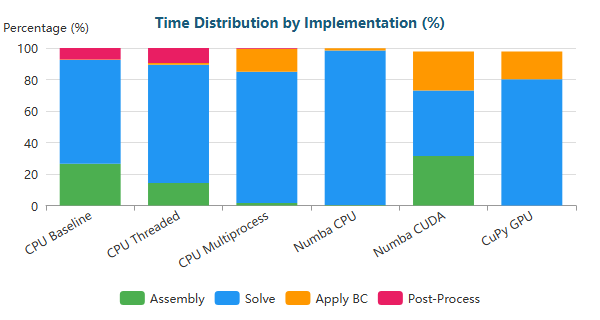
**Time Distribution:**



Pie



Pie

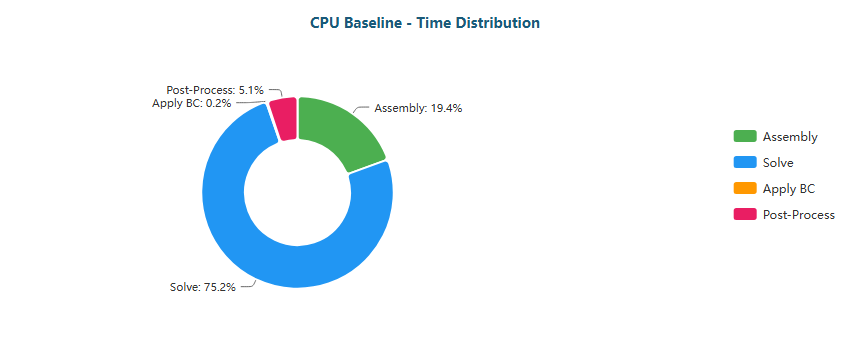


Stacked Bar

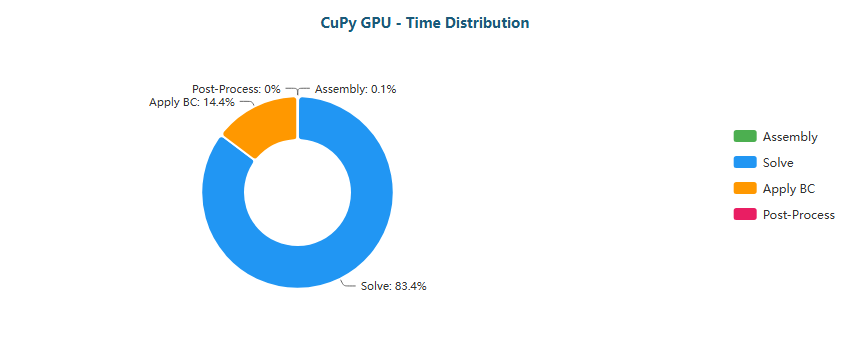
#### Y-Shaped (XL) - 1,357,953 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (75%) | Assembly (19%) |
| CPU Threaded | Solve (83%) | Assembly (10%) |
| CPU Multiprocess | Solve (87%) | BC (12%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (41%) | Assembly (30%) |
| CuPy GPU | Solve (83%) | BC (14%) |

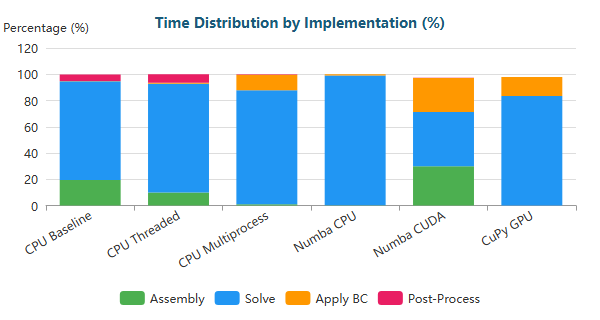
**Time Distribution:**



Pie



Pie



Stacked Bar

### Why Each Optimization Helps

| Transition | Reason |
| --- | --- |
| Baseline → Threaded | Limited by Python GIL; threads only help for I/O |
| Threaded → Multiprocess | Bypasses GIL via separate processes; IPC overhead limits gains |
| Multiprocess → Numba CPU | JIT compilation eliminates interpreter overhead; true parallel loops |
| Numba CPU → Numba CUDA | GPU parallelism: thousands of threads vs dozens of CPU cores |
| Numba CUDA → CuPy GPU | CUDA C kernels more optimized than Numba-generated PTX |

#### 1.2.2.1 Bottleneck Migration on RTX 4090

From the RTX 4090 profiling, the bottleneck migration follows the same macro-pattern observed in higher-tier GPUs, but with clearer evidence of **fixed GPU overheads** at small scale and a more visible **solver/BC trade-off** in CuPy for XS meshes:

* **CPU Baseline → CPU Threaded**
  + The pipeline remained **assembly-dominated** in XS meshes (≈69% → 52% Assembly), with Post-Processing still relevant (≈20–24%).
  + Threads reduced wall-time but did not fundamentally change the bottleneck structure, confirming that the workload is not fully thread-scalable at Python level.
* **CPU Multiprocess**
  + Unlike the RTX 5090 case where Post-Processing could dominate, here **Assembly stayed high even in multiprocess** (≈50–53% in XS cases), while Post-Processing rose sharply (≈45–47%).
  + This suggests that **IPC + process orchestration overheads** compete with actual computation and can “freeze” the bottleneck at high-level stages.
* **Numba CPU (JIT)**
  + The bottleneck shifted decisively to **Solve**, especially from M upward:
    - M/L/XL: Solve ≈98–99% (with residual BC ≈0–1%).
  + This indicates that once interpreter overhead is eliminated, **numerical linear algebra becomes the true limiting factor** even on CPU.
* **Numba CUDA**
  + For XS: Solve became overwhelming (≈85–91%), reflecting that GPU execution is dominated by the solver kernel *relative* cost once assembly is minimized.
  + For M/L/XL: a consistent split emerged where **Assembly resurfaces as a significant share** (≈30–37%), while Solve remained ≈40–53%.
  + Interpretation: on RTX 4090, GPU acceleration makes both stages fast, but the *relative balance* becomes sensitive to how assembly is implemented (kernel fusion, memory writes, launch structure).
* **CuPy GPU**
  + For M/L/XL: CuPy converged to the classic GPU sparse profile:
    - Solve ≈83–91% (dominant), BC ≈9–16% (secondary).
  + For XS: Solve still dominated (≈60–73%), but **BC became unusually visible** (≈22–30%).
  + Key implication: on small meshes, **kernel launch/synchronization + BC handling overheads** are not amortized.

Overall, the RTX 4090 results confirm that once GPU acceleration is active, **assembly and post-processing rapidly collapse in relative importance**, and the pipeline becomes dominated by **sparse solver behavior**, with **BC application** acting as a persistent secondary limiter—especially at small scale.

#### 1.2.2.2 Optimization Implications and Limits on RTX 4090

The RTX 4090 bottleneck structure points to a clear hierarchy of where additional speedups can still be extracted:

* **Assembly improvements are only impactful in mid-scale GPU runs**
  + In Numba CUDA for M/L/XL, assembly still contributes ≈30–37% of runtime.
  + Practical direction: reduce kernel launches, fuse element operations, minimize scatter writes, and improve memory coalescing.
* **CuPy performance is solver-bound by design**
  + In CuPy GPU, the solver dominates at all meaningful scales (≈83–91%).
  + Therefore, further gains require **algorithmic improvements**, not kernel micro-optimizations:
    - Reduce iteration count (better preconditioning, improved conditioning).
    - Use more suitable solvers (e.g., AMG-style preconditioners, or alternative Krylov methods depending on matrix properties).
    - Improve sparse format choices (CSR/ELL/HYB) depending on sparsity pattern.
* **Boundary conditions become a structural tax on GPU**
  + BC is consistently the **second bottleneck** in CuPy GPU (≈9–19% for M–XL, ≈22–30% for XS).
  + Practical direction:
    - Apply BC through mask-based operations with fewer sync points.
    - Avoid repeated global memory passes.
    - If feasible, fold BC enforcement into solver iterations or pre-processing steps.
* **Small meshes remain CPU-favorable**
  + XS cases show that fixed GPU overheads (launch/sync/dispatch) and BC handling prevent strong GPU gains.
  + Best practice remains:
    - **XS meshes:** Numba JIT CPU (lowest overhead, compiled execution).
    - **M–XL meshes:** CuPy GPU (solver-dominant, best overall scaling).

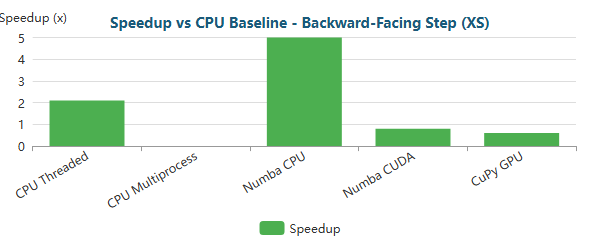
The RTX 4090 exposes a transition where performance is constrained less by “raw compute” and more by **sparse memory efficiency** and **pipeline-level design choices**, with the solver and BC enforcement defining the practical ceiling for further acceleration.

### 1.3 RTX 5070 Performance

Key results from performance benchmarks comparing FEM solver implementations.

**Backward-Facing Step (XS)** (287 nodes)

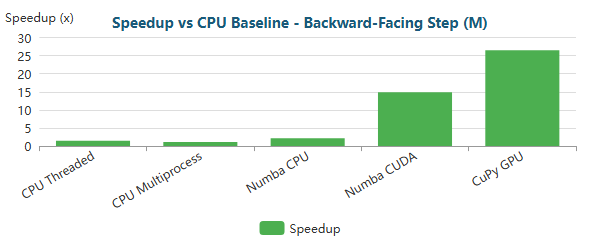
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 30ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 14ms ± 0ms | 2.1x | 3 |
| CPU Multiprocess | 3.09s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 5.0x | 3 |
| Numba CUDA | 39ms ± 0ms | 0.8x | 3 |
| CuPy GPU | 49ms ± 2ms | 0.6x | 3 |



Bar

**Backward-Facing Step (M)** (195,362 nodes)

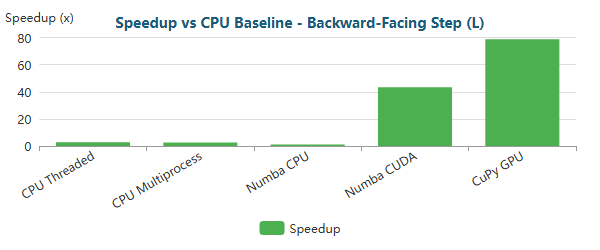
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 38.03s ± 0.13s | 1.0x | 3 |
| CPU Threaded | 24.86s ± 0.08s | 1.5x | 3 |
| CPU Multiprocess | 34.59s ± 0.35s | 1.1x | 3 |
| Numba CPU | 17.04s ± 0.08s | 2.2x | 3 |
| Numba CUDA | 2.56s ± 0.03s | 14.9x | 3 |
| CuPy GPU | 1.43s ± 0.03s | 26.5x | 3 |



Bar

**Backward-Facing Step (L)** (766,088 nodes)

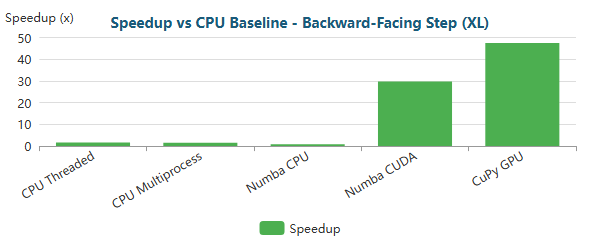
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 8m 12.1s ± 0.6s | 1.0x | 3 |
| CPU Threaded | 2m 55.0s ± 0.3s | 2.8x | 3 |
| CPU Multiprocess | 3m 8.8s ± 0.3s | 2.6x | 3 |
| Numba CPU | 7m 9.0s ± 0.1s | 1.1x | 3 |
| Numba CUDA | 11.36s ± 0.19s | 43.3x | 3 |
| CuPy GPU | 6.25s ± 0.01s | 78.7x | 3 |



Bar

**Backward-Facing Step (XL)** (1,283,215 nodes)

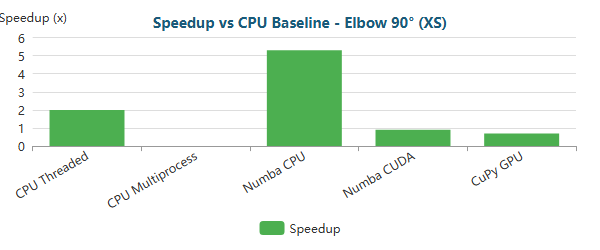
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 10m 23.6s ± 2.4s | 1.0x | 3 |
| CPU Threaded | 6m 37.8s ± 0.5s | 1.6x | 3 |
| CPU Multiprocess | 7m 6.1s ± 1.7s | 1.5x | 3 |
| Numba CPU | 11m 20.1s ± 168.0s | 0.9x | 3 |
| Numba CUDA | 20.97s ± 0.06s | 29.7x | 3 |
| CuPy GPU | 13.13s ± 0.03s | 47.5x | 3 |



Bar

**Elbow 90° (XS)** (411 nodes)

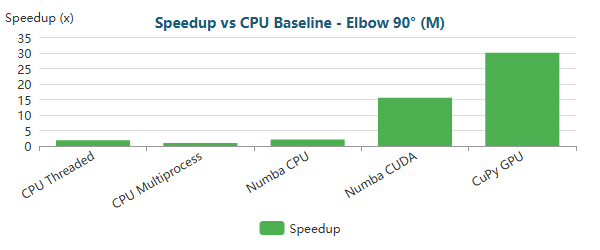
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 39ms ± 0ms | 1.0x | 3 |
| CPU Threaded | 19ms ± 0ms | 2.0x | 3 |
| CPU Multiprocess | 3.86s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 5.3x | 3 |
| Numba CUDA | 44ms ± 3ms | 0.9x | 3 |
| CuPy GPU | 53ms ± 4ms | 0.7x | 3 |



Bar

**Elbow 90° (M)** (161,984 nodes)

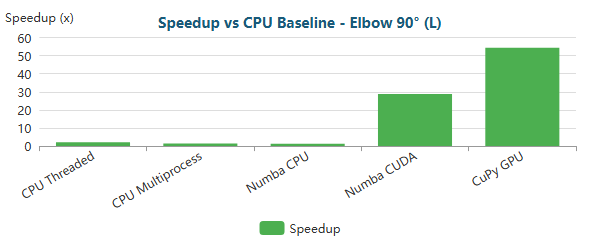
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 32.80s ± 0.17s | 1.0x | 3 |
| CPU Threaded | 17.39s ± 0.05s | 1.9x | 3 |
| CPU Multiprocess | 33.69s ± 0.24s | 1.0x | 3 |
| Numba CPU | 15.58s ± 0.05s | 2.1x | 3 |
| Numba CUDA | 2.10s ± 0.05s | 15.6x | 3 |
| CuPy GPU | 1.09s ± 0.01s | 30.1x | 3 |



Bar

**Elbow 90° (L)** (623,153 nodes)

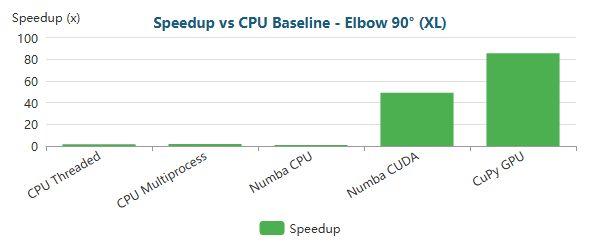
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 3m 54.8s ± 0.5s | 1.0x | 3 |
| CPU Threaded | 1m 52.8s ± 0.3s | 2.1x | 3 |
| CPU Multiprocess | 2m 50.5s ± 0.5s | 1.4x | 3 |
| Numba CPU | 2m 54.1s ± 3.3s | 1.3x | 3 |
| Numba CUDA | 8.15s ± 0.23s | 28.8x | 3 |
| CuPy GPU | 4.32s ± 0.03s | 54.3x | 3 |



Bar

**Elbow 90° (XL)** (1,044,857 nodes)

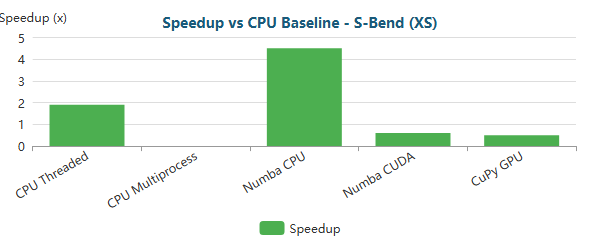
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 12m 4.5s ± 5.4s | 1.0x | 3 |
| CPU Threaded | 6m 34.2s ± 128.6s | 1.8x | 3 |
| CPU Multiprocess | 6m 4.8s ± 3.5s | 2.0x | 3 |
| Numba CPU | 11m 8.7s ± 0.8s | 1.1x | 3 |
| Numba CUDA | 14.73s ± 0.11s | 49.2x | 3 |
| CuPy GPU | 8.47s ± 0.06s | 85.6x | 3 |



Bar

**S-Bend (XS)** (387 nodes)

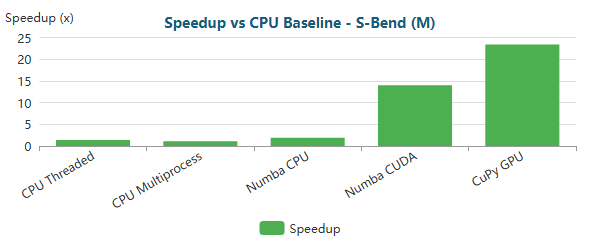
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 31ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 16ms ± 0ms | 1.9x | 3 |
| CPU Multiprocess | 3.07s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 4.5x | 3 |
| Numba CUDA | 48ms ± 2ms | 0.6x | 3 |
| CuPy GPU | 60ms ± 1ms | 0.5x | 3 |



Bar

**S-Bend (M)** (196,078 nodes)

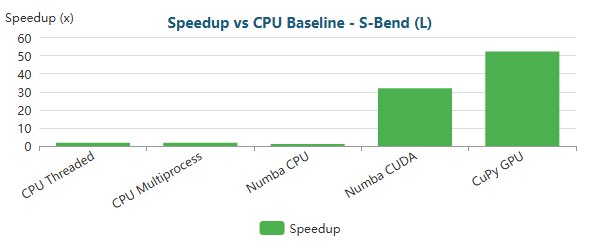
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 38.78s ± 0.12s | 1.0x | 3 |
| CPU Threaded | 26.99s ± 0.02s | 1.4x | 3 |
| CPU Multiprocess | 35.01s ± 0.02s | 1.1x | 3 |
| Numba CPU | 20.32s ± 0.32s | 1.9x | 3 |
| Numba CUDA | 2.78s ± 0.02s | 14.0x | 3 |
| CuPy GPU | 1.66s ± 0.01s | 23.4x | 3 |



Bar

**S-Bend (L)** (765,441 nodes)

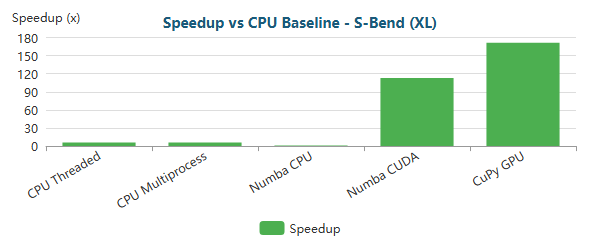
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 18.7s ± 0.1s | 1.0x | 3 |
| CPU Threaded | 3m 16.0s ± 0.6s | 1.9x | 3 |
| CPU Multiprocess | 3m 15.9s ± 0.5s | 1.9x | 3 |
| Numba CPU | 5m 13.9s ± 0.1s | 1.2x | 3 |
| Numba CUDA | 11.85s ± 0.14s | 31.9x | 3 |
| CuPy GPU | 7.25s ± 0.02s | 52.3x | 3 |



Bar

**S-Bend (XL)** (1,286,039 nodes)

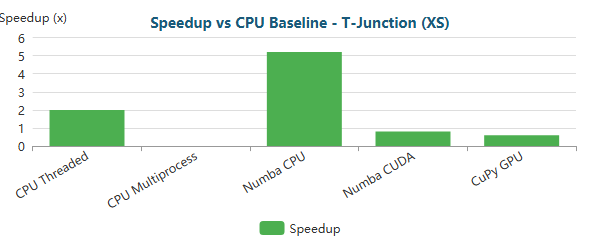
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 45m 10.2s ± 2.6s | 1.0x | 3 |
| CPU Threaded | 7m 47.2s ± 5.7s | 5.8x | 3 |
| CPU Multiprocess | 7m 33.9s ± 12.7s | 6.0x | 3 |
| Numba CPU | 52m 16.5s ± 58.7s | 0.9x | 3 |
| Numba CUDA | 24.04s ± 0.12s | 112.8x | 3 |
| CuPy GPU | 15.82s ± 0.12s | 171.3x | 3 |



Bar

**T-Junction (XS)** (393 nodes)

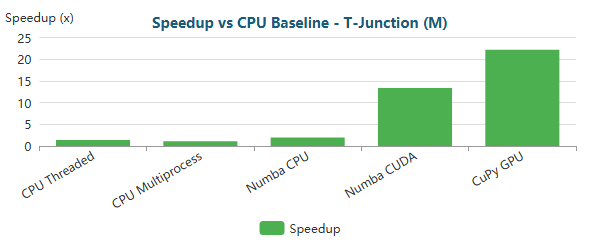
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 39ms ± 0ms | 1.0x | 3 |
| CPU Threaded | 19ms ± 0ms | 2.0x | 3 |
| CPU Multiprocess | 3.90s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 2ms | 5.2x | 3 |
| Numba CUDA | 49ms ± 1ms | 0.8x | 3 |
| CuPy GPU | 69ms ± 8ms | 0.6x | 3 |



Bar

**T-Junction (M)** (196,420 nodes)

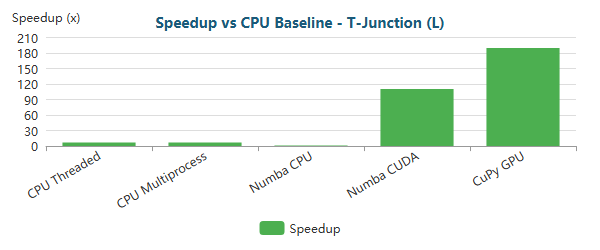
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 36.05s ± 0.18s | 1.0x | 3 |
| CPU Threaded | 26.06s ± 0.08s | 1.4x | 3 |
| CPU Multiprocess | 34.19s ± 0.12s | 1.1x | 3 |
| Numba CPU | 18.20s ± 0.12s | 2.0x | 3 |
| Numba CUDA | 2.69s ± 0.01s | 13.4x | 3 |
| CuPy GPU | 1.62s ± 0.03s | 22.2x | 3 |



Bar

**T-Junction (L)** (768,898 nodes)

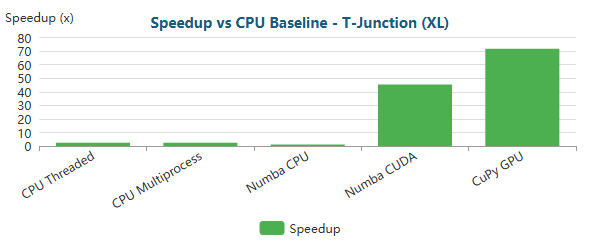
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 21m 41.1s ± 1.2s | 1.0x | 3 |
| CPU Threaded | 3m 4.9s ± 0.6s | 7.0x | 3 |
| CPU Multiprocess | 3m 5.7s ± 0.3s | 7.0x | 3 |
| Numba CPU | 18m 42.9s ± 6.0s | 1.2x | 3 |
| Numba CUDA | 11.79s ± 0.01s | 110.4x | 3 |
| CuPy GPU | 6.87s ± 0.16s | 189.5x | 3 |



Bar

**T-Junction (XL)** (1,291,289 nodes)

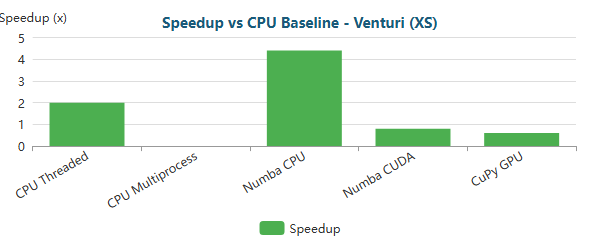
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 16m 45.0s ± 4.5s | 1.0x | 3 |
| CPU Threaded | 7m 6.6s ± 1.1s | 2.4x | 3 |
| CPU Multiprocess | 7m 5.2s ± 1.8s | 2.4x | 3 |
| Numba CPU | 14m 21.0s ± 2.7s | 1.2x | 3 |
| Numba CUDA | 22.16s ± 0.14s | 45.3x | 3 |
| CuPy GPU | 14.02s ± 0.01s | 71.7x | 3 |



Bar

**Venturi (XS)** (341 nodes)

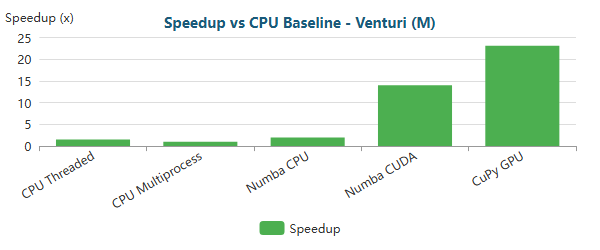
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 32ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 16ms ± 1ms | 2.0x | 3 |
| CPU Multiprocess | 3.09s ± 0.02s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 4.4x | 3 |
| Numba CUDA | 40ms ± 2ms | 0.8x | 3 |
| CuPy GPU | 52ms ± 2ms | 0.6x | 3 |



Bar

**Venturi (M)** (194,325 nodes)

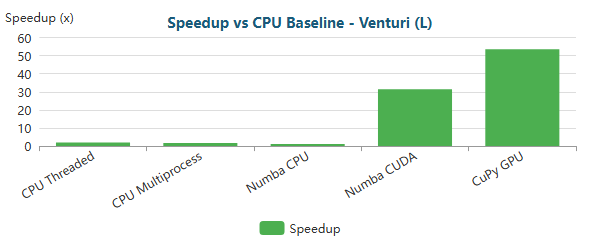
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 35.94s ± 0.07s | 1.0x | 3 |
| CPU Threaded | 24.67s ± 0.16s | 1.5x | 3 |
| CPU Multiprocess | 35.95s ± 0.45s | 1.0x | 3 |
| Numba CPU | 17.61s ± 0.03s | 2.0x | 3 |
| Numba CUDA | 2.56s ± 0.04s | 14.0x | 3 |
| CuPy GPU | 1.55s ± 0.04s | 23.1x | 3 |



Bar

**Venturi (L)** (763,707 nodes)

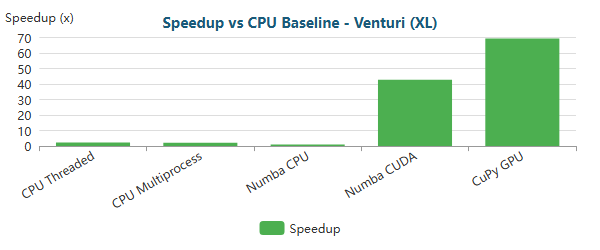
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 5m 36.6s ± 0.1s | 1.0x | 3 |
| CPU Threaded | 2m 48.3s ± 0.4s | 2.0x | 3 |
| CPU Multiprocess | 3m 14.5s ± 0.6s | 1.7x | 3 |
| Numba CPU | 4m 35.5s ± 2.2s | 1.2x | 3 |
| Numba CUDA | 10.71s ± 0.14s | 31.4x | 3 |
| CuPy GPU | 6.30s ± 0.02s | 53.5x | 3 |



Bar

**Venturi (XL)** (1,284,412 nodes)

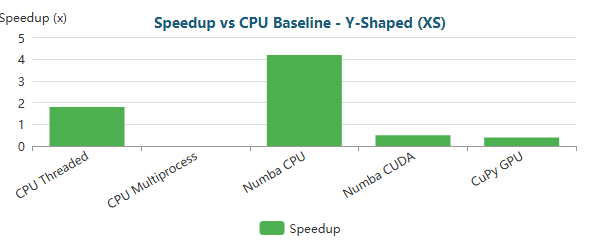
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 14m 46.7s ± 2.5s | 1.0x | 3 |
| CPU Threaded | 6m 18.8s ± 1.3s | 2.3x | 3 |
| CPU Multiprocess | 7m 9.8s ± 1.0s | 2.1x | 3 |
| Numba CPU | 13m 7.4s ± 7.7s | 1.1x | 3 |
| Numba CUDA | 20.79s ± 0.06s | 42.7x | 3 |
| CuPy GPU | 12.79s ± 0.05s | 69.3x | 3 |



Bar

**Y-Shaped (XS)** (201 nodes)

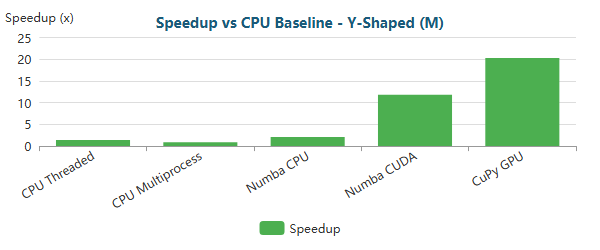
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 19ms ± 0ms | 1.0x | 3 |
| CPU Threaded | 11ms ± 0ms | 1.8x | 3 |
| CPU Multiprocess | 2.34s ± 0.02s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 4.2x | 3 |
| Numba CUDA | 40ms ± 4ms | 0.5x | 3 |
| CuPy GPU | 48ms ± 2ms | 0.4x | 3 |



Bar

**Y-Shaped (M)** (195,853 nodes)

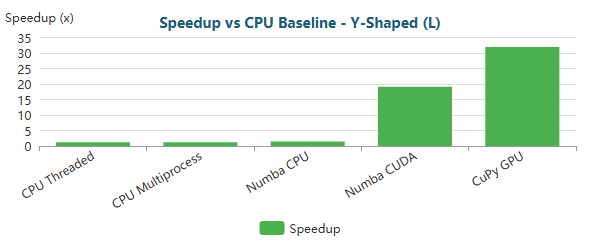
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 28.02s ± 0.07s | 1.0x | 3 |
| CPU Threaded | 20.52s ± 0.12s | 1.4x | 3 |
| CPU Multiprocess | 29.85s ± 0.04s | 0.9x | 3 |
| Numba CPU | 13.28s ± 0.08s | 2.1x | 3 |
| Numba CUDA | 2.38s ± 0.10s | 11.8x | 3 |
| CuPy GPU | 1.38s ± 0.01s | 20.3x | 3 |



Bar

**Y-Shaped (L)** (772,069 nodes)

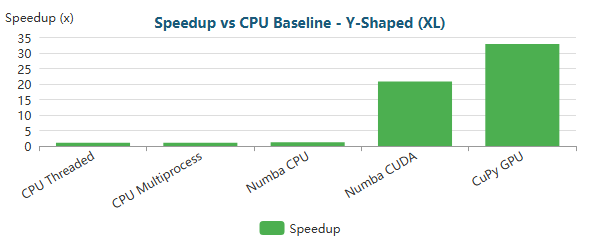
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 3m 2.7s ± 2.4s | 1.0x | 3 |
| CPU Threaded | 2m 33.6s ± 0.2s | 1.2x | 3 |
| CPU Multiprocess | 2m 36.3s ± 1.5s | 1.2x | 3 |
| Numba CPU | 2m 1.2s ± 2.9s | 1.5x | 3 |
| Numba CUDA | 9.58s ± 0.18s | 19.1x | 3 |
| CuPy GPU | 5.72s ± 0.02s | 31.9x | 3 |



Bar

**Y-Shaped (XL)** (1,357,953 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 42.8s ± 1.0s | 1.0x | 3 |
| CPU Threaded | 6m 16.5s ± 0.9s | 1.1x | 3 |
| CPU Multiprocess | 6m 4.3s ± 1.1s | 1.1x | 3 |
| Numba CPU | 5m 34.5s ± 2.2s | 1.2x | 3 |
| Numba CUDA | 19.35s ± 0.09s | 20.8x | 3 |
| CuPy GPU | 12.25s ± 0.05s | 32.9x | 3 |



Bar

### 1.3.1 Critical Analysis RTX 4070

#### 1.3.1.1 Small-Scale Problems (XS meshes)

Across all geometries with **XS meshes (≈200–400 nodes)**, the results show a **clear overhead-dominated regime**:

* **CPU-based implementations dominate** absolute performance.
* **Numba JIT CPU** is consistently the fastest option, with speedups typically around **~4.2× to ~5.3×** relative to the CPU baseline (e.g., **5.0×** Backward-Facing Step, **5.3×** Elbow 90°, **4.5×** S-Bend, **5.2×** T-Junction, **4.4×** Venturi, **4.2×** Y-shaped).
* **GPU implementations (Numba CUDA and CuPy GPU)** are systematically *slower* than the CPU baseline, with speedups in the range **~0.4× to ~0.9×**, reflecting real slowdowns.

This behavior is expected. At XS scale, FEM runtime is dominated by fixed overheads (kernel launch latency, device synchronization, and GPU runtime management), which cannot be amortized with only a few hundred nodes. As a result, even though the GPU can execute kernels quickly, end-to-end execution remains overhead-limited, confirming that **GPU acceleration is not suitable for small FEM problems**.

Additionally, **CPU multiprocessing performs extremely poorly** in this regime (often several seconds, i.e., ~0× speedup), as process startup and IPC costs dominate execution time. Threading helps (≈1.8×–2.1×), but still trails behind JIT compilation.

#### 1.3.1.2 Medium-Scale Problems (M meshes)

For **medium-scale meshes (≈160k–200k nodes)**, the performance profile shifts consistently across all geometries, marking a clear **CPU–GPU crossover**:

* **GPU acceleration becomes dominant** and stable.
* **Numba CUDA** reaches speedups of approximately **~11.8× to ~15.6×** (e.g., **11.8×** Y-shaped, **13.4×** T-junction, **14.0×** S-bend/Venturi, **14.9×** backward-facing step, **15.6×** elbow).
* **CuPy GPU (RawKernel)** consistently outperforms Numba CUDA, reaching approximately **~20× to ~30×** (e.g., **20.3×** Y-shaped, **22.2×** T-junction, **23.1×** Venturi, **23.4×** S-bend, **26.5×** backward-facing step, **30.1×** elbow).
* **CPU-based approaches saturate**, typically capped around **~1.4×–2.2×**, even with threading or JIT, confirming that CPU improvements become marginal once the workload is large.

At this scale, there is enough parallel work to saturate the GPU, and assembly/post-processing become relatively cheap. Total runtime increasingly reflects solver behavior, while CuPy’s lower abstraction overhead and stronger GPU residency enable consistently higher speedups than Numba CUDA.

#### 1.3.1.3 Large-Scale and Extreme-Scale Problems (L and XL meshes)

For **large (L) and extra-large (XL) meshes (≈600k–1.35M nodes)**, GPU acceleration becomes **essential**:

* **CPU baseline runtimes grow to minutes** across all geometries (and up to ~45 minutes for S-Bend XL), making CPU-only execution impractical at scale.
* **Threading and multiprocessing provide limited relief**, typically around **~1.1×–2.8×**, with a notable higher gain in **S-Bend XL (~5.8×–6.0×)** but still far below GPU performance.
* **Numba JIT CPU loses effectiveness**, often approaching baseline (≈0.9×–1.5×), consistent with a memory-bandwidth-bound sparse regime.
* **Numba CUDA achieves strong speedups**, ranging approximately **~19×–110×** for L and **~20×–112×** for XL depending on geometry.
* **CuPy GPU defines the performance envelope**, reaching **~32×–190×** speedup for the largest cases (e.g., **189.5×** for T-Junction L, **171.3×** for S-Bend XL).

At these scales, assembly and post-processing are effectively amortized, and the runtime is dominated by the sparse iterative solver. Speedups flatten as the solver becomes **memory-bandwidth bound** with sparse/irregular access patterns, limiting how far hardware capability alone can push performance.

#### 1.3.1.4 Comparative Assessment Across Scales

From a practical standpoint, the results support the following execution-model selection:

* **XS meshes:** Numba JIT CPU — minimal overhead and compiled execution.
* **M meshes:** CuPy GPU (RawKernel) — best CPU–GPU crossover efficiency and consistent throughput.
* **L and XL meshes:** CuPy GPU (RawKernel) — maximum scalability and best end-to-end runtime.
* **CPU-only environments:** Numba JIT CPU — best balance of speed and portability.
* **GPU prototyping:** Numba CUDA — faster development with solid speedups.
* **Production GPU workloads:** CuPy RawKernel — highest and most consistent speedups.

#### 1.3.1.5 Comparative Assessment of Execution Models

From a practical standpoint, the benchmark results support the following conclusions:

| Regime | Best Execution Model | Rationale |
| --- | --- | --- |
| XS meshes | Numba JIT CPU | Minimal overhead, compiled execution |
| M meshes | CuPy GPU (RawKernel) | Maximum throughput, full GPU residency |
| CPU-only environments | Numba JIT CPU | Best balance of speed and memory efficiency |
| GPU prototyping | Numba CUDA | Easier development, acceptable performance |
| Production GPU | CuPy RawKernel | Highest and most consistent speedups |

The RTX 4070 demonstrates **clear scalability once problem size justifies GPU usage**, but the results also highlight that **problem scale and solver behavior dominate performance**. Overall, the benchmarks confirm that the RTX 4070 can deliver **order-of-magnitude speedups** for medium to extreme FEM workloads when GPU execution is implemented efficiently—especially with **RawKernel-based CuPy**. At the same time, the data reinforces critical best practices:

* GPU acceleration should be **selectively applied**, not used indiscriminately.
* Small and interactive FEM problems are better served by optimized CPU execution (Numba JIT CPU).
* For large-scale workloads, **RawKernel GPU implementations provide the highest return**.
* At scale, the **sparse linear solver—not the assembly kernel—becomes the dominant bottleneck**, and further gains depend on solver-level improvements rather than faster kernels alone.

Taken together, these results position the RTX 4070 as a capable GPU for FEM acceleration at realistic scales, while confirming that sparse linear algebra efficiency ultimately defines the performance ceiling.

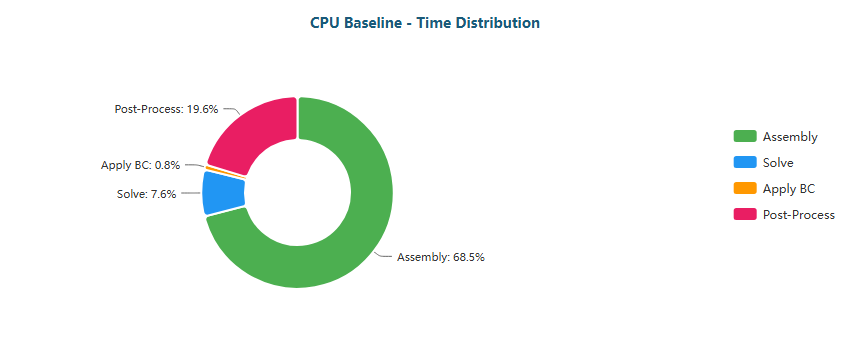
### 1.3.2 RTX 4070 Bottleneck Evolution Critical Analysis

As optimizations progress, the computational bottleneck shifts:

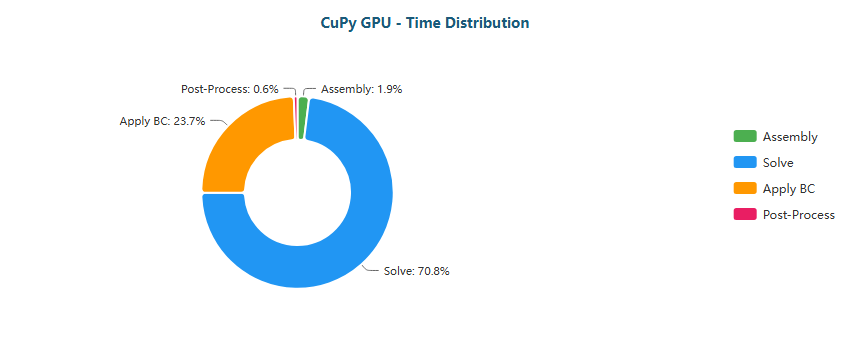
#### Backward-Facing Step (XS) - 287 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (68%) | Post-Proc (20%) |
| CPU Threaded | Assembly (53%) | Post-Proc (23%) |
| CPU Multiprocess | Post-Proc (62%) | Assembly (38%) |
| Numba CPU | Solve (53%) | BC (14%) |
| Numba CUDA | Solve (86%) | Assembly (7%) |
| CuPy GPU | Solve (71%) | BC (24%) |

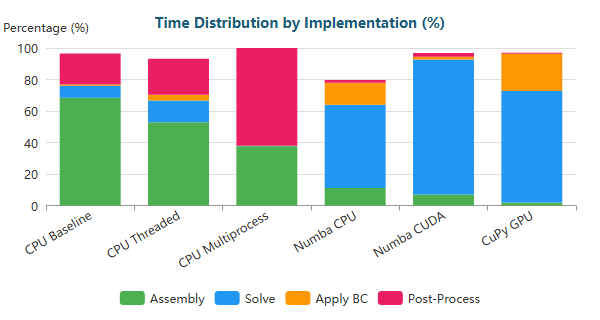
**Time Distribution:**



Pie



Pie

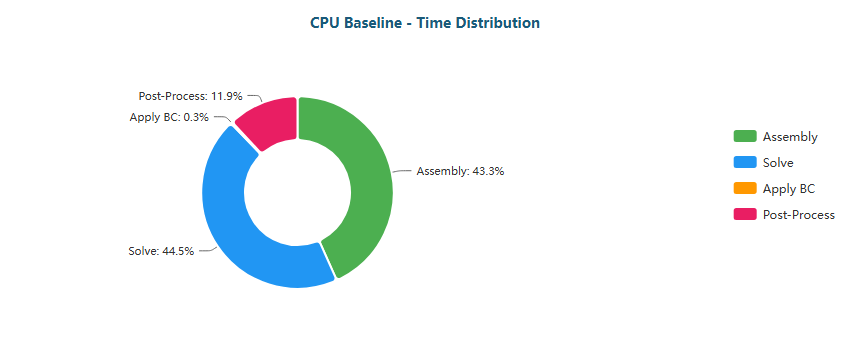


Stacked Bar

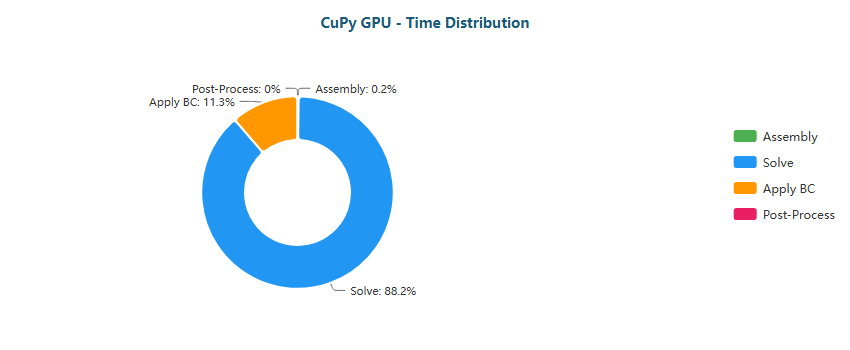
#### Backward-Facing Step (M) - 195,362 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (45%) | Assembly (43%) |
| CPU Threaded | Solve (62%) | Assembly (25%) |
| CPU Multiprocess | Solve (42%) | Assembly (21%) |
| Numba CPU | Solve (96%) | BC (3%) |
| Numba CUDA | Solve (48%) | Assembly (32%) |
| CuPy GPU | Solve (88%) | BC (11%) |

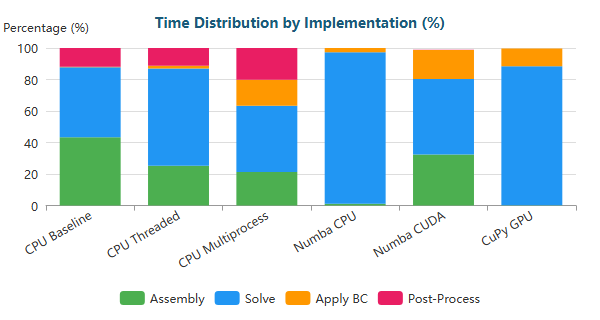
**Time Distribution:**



Pie



Pie

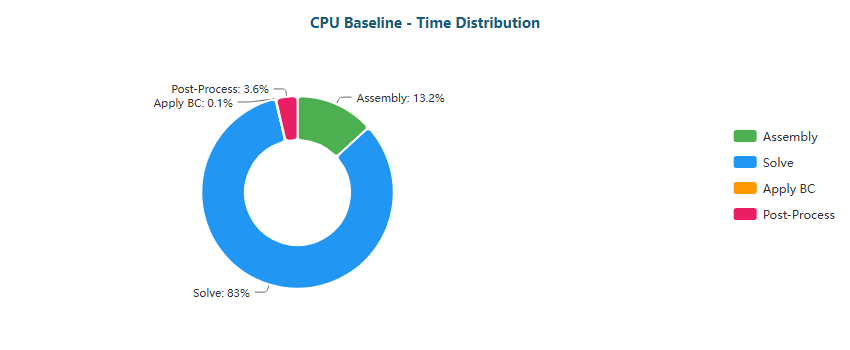


Stacked Bar

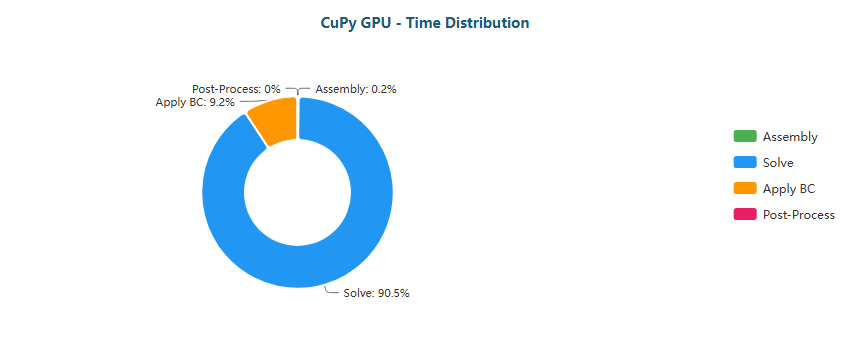
#### Backward-Facing Step (L) - 766,088 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (83%) | Assembly (13%) |
| CPU Threaded | Solve (79%) | Assembly (14%) |
| CPU Multiprocess | Solve (70%) | BC (21%) |
| Numba CPU | Solve (99%) | BC (0%) |
| Numba CUDA | Solve (53%) | Assembly (29%) |
| CuPy GPU | Solve (90%) | BC (9%) |

**Time Distribution:**



Pie



Pie

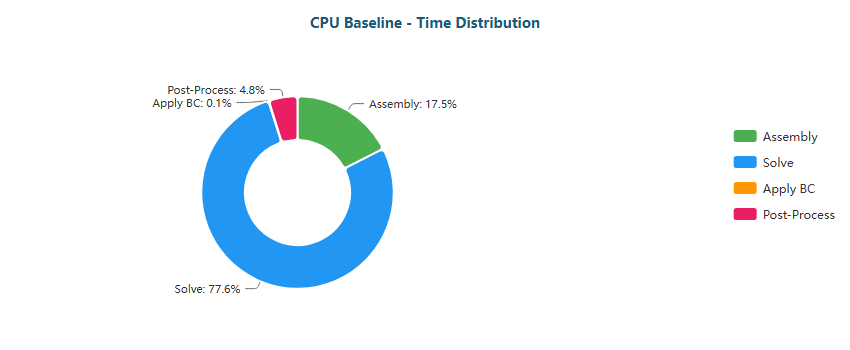


Stacked Bar

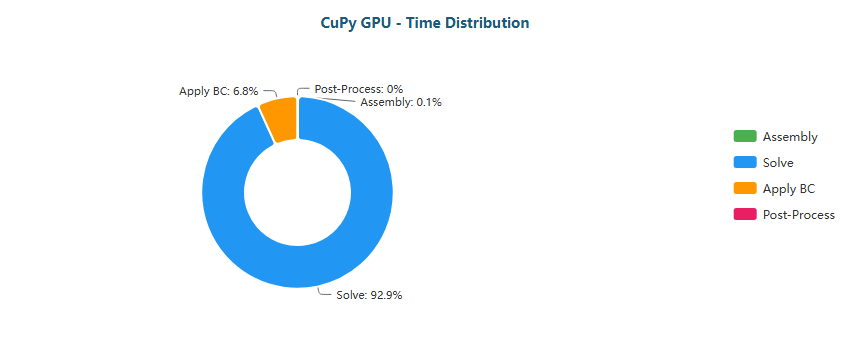
#### Backward-Facing Step (XL) - 1,283,215 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (78%) | Assembly (17%) |
| CPU Threaded | Solve (85%) | Assembly (10%) |
| CPU Multiprocess | Solve (75%) | BC (21%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (58%) | Assembly (26%) |
| CuPy GPU | Solve (93%) | BC (7%) |

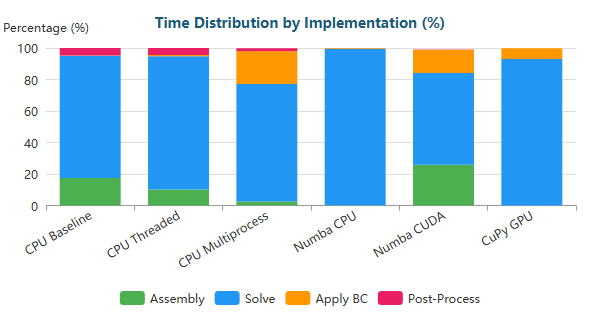
**Time Distribution:**



Pie



Pie

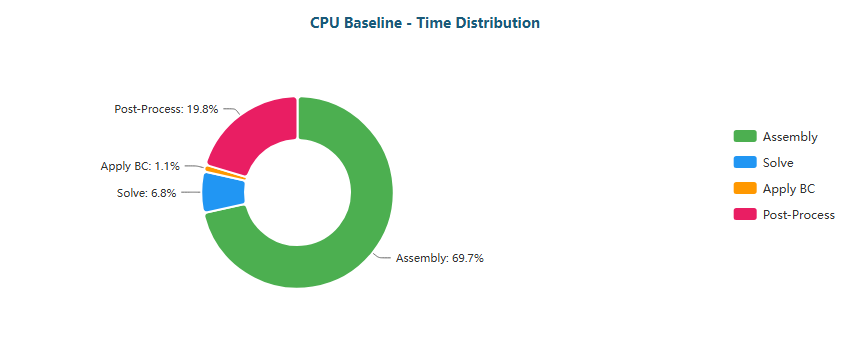


Stacked Bar

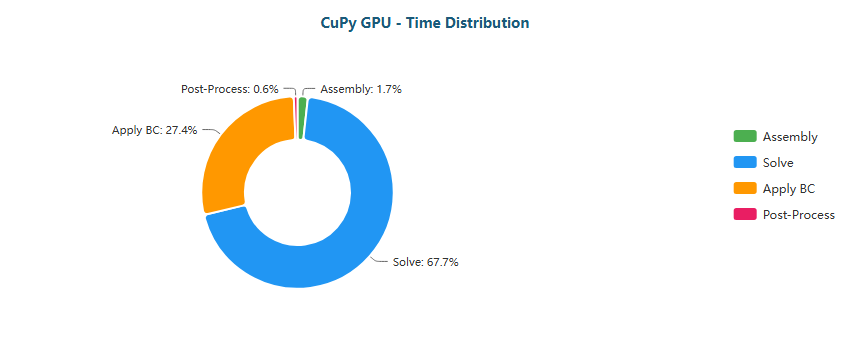
#### Elbow 90° (XS) - 411 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (70%) | Post-Proc (20%) |
| CPU Threaded | Assembly (54%) | Post-Proc (23%) |
| CPU Multiprocess | Post-Proc (50%) | Assembly (50%) |
| Numba CPU | Solve (48%) | BC (19%) |
| Numba CUDA | Solve (85%) | Assembly (8%) |
| CuPy GPU | Solve (68%) | BC (27%) |

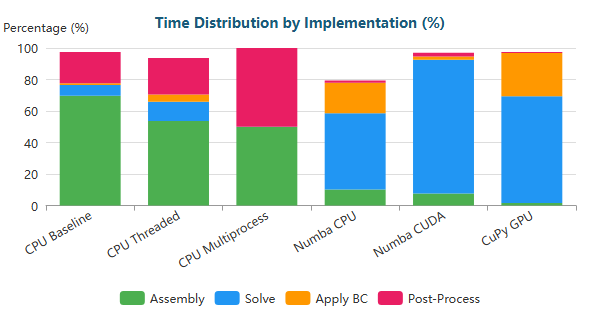
**Time Distribution:**



Pie



Pie

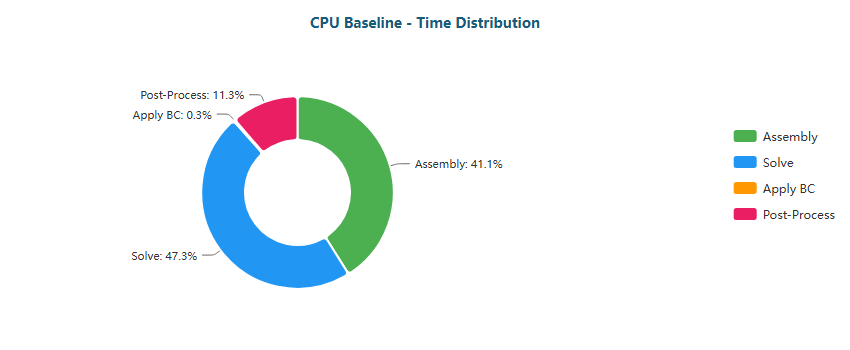


Stacked Bar

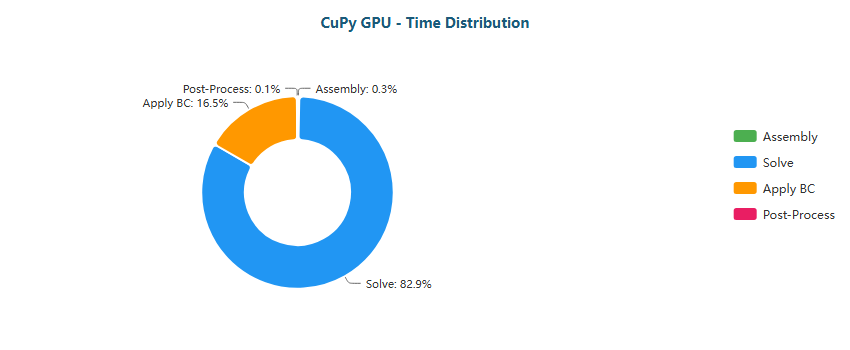
#### Elbow 90° (M) - 161,984 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (47%) | Assembly (41%) |
| CPU Threaded | Solve (57%) | Assembly (29%) |
| CPU Multiprocess | BC (30%) | Solve (28%) |
| Numba CPU | Solve (96%) | BC (2%) |
| Numba CUDA | Solve (48%) | Assembly (33%) |
| CuPy GPU | Solve (83%) | BC (17%) |

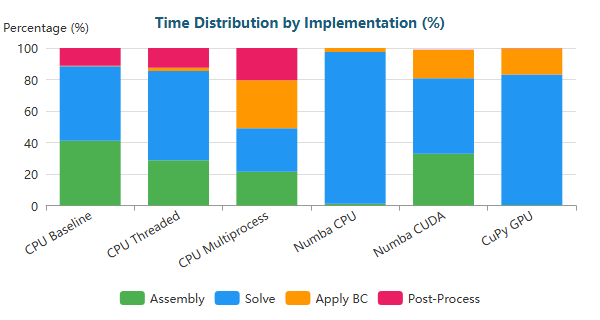
**Time Distribution:**



Pie



Pie

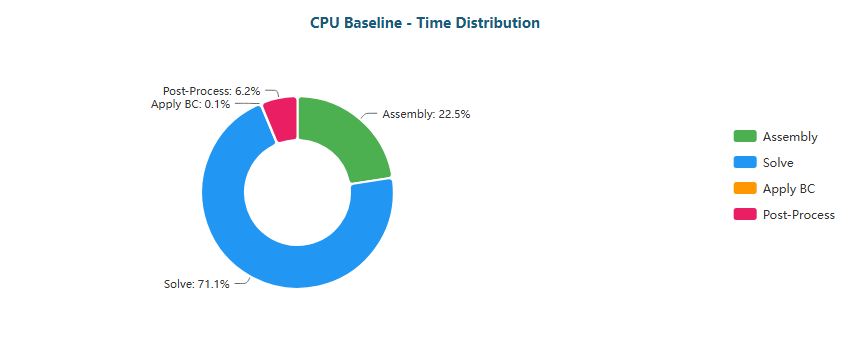


Stacked Bar

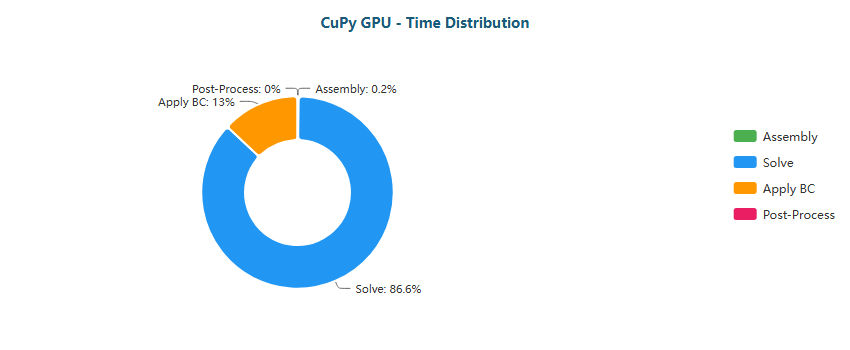
#### Elbow 90° (L) - 623,153 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (71%) | Assembly (23%) |
| CPU Threaded | Solve (73%) | Assembly (17%) |
| CPU Multiprocess | Solve (47%) | BC (44%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (48%) | Assembly (33%) |
| CuPy GPU | Solve (87%) | BC (13%) |

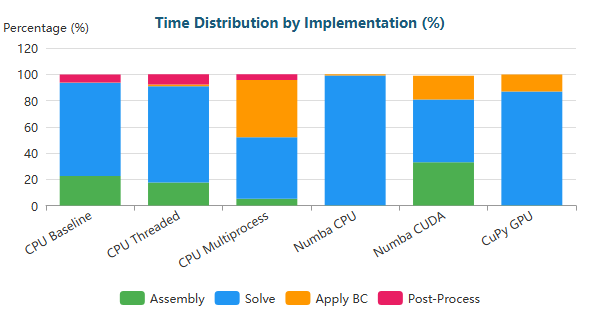
**Time Distribution:**



Pie



Pie

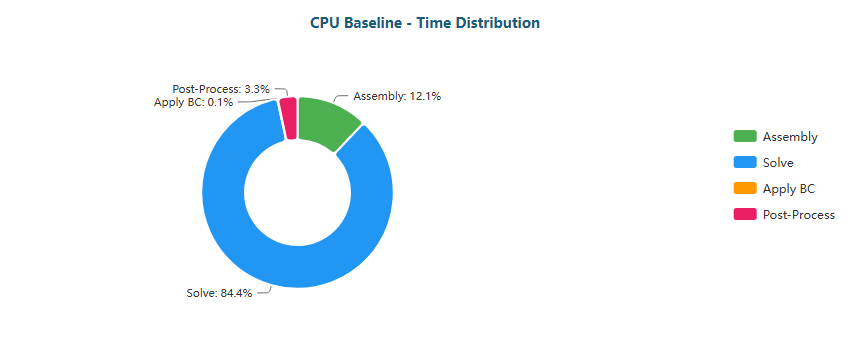


Stacked Bar

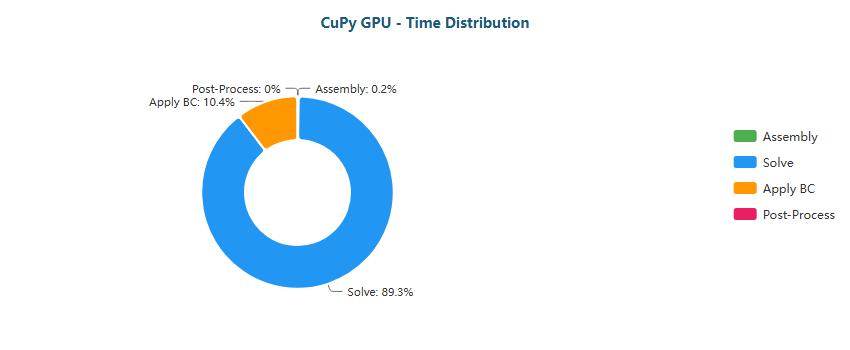
#### Elbow 90° (XL) - 1,044,857 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (84%) | Assembly (12%) |
| CPU Threaded | Solve (81%) | Assembly (12%) |
| CPU Multiprocess | Solve (51%) | BC (44%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (52%) | Assembly (30%) |
| CuPy GPU | Solve (89%) | BC (10%) |

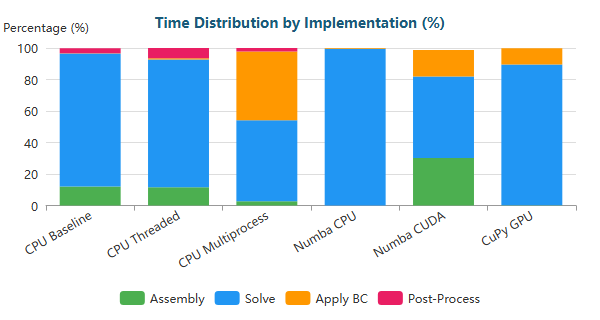
**Time Distribution:**



Pie



Pie

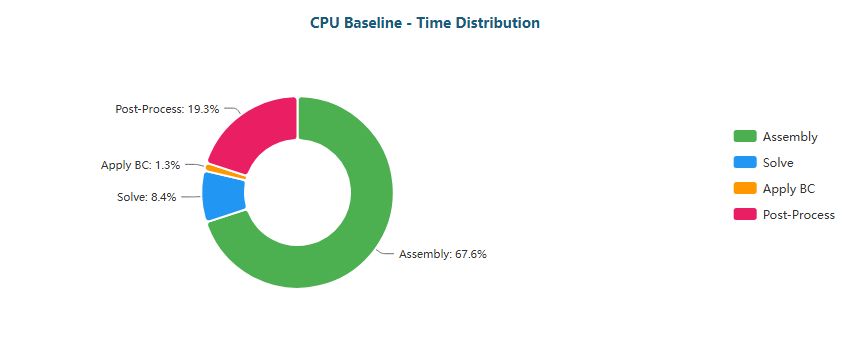


Stacked Bar

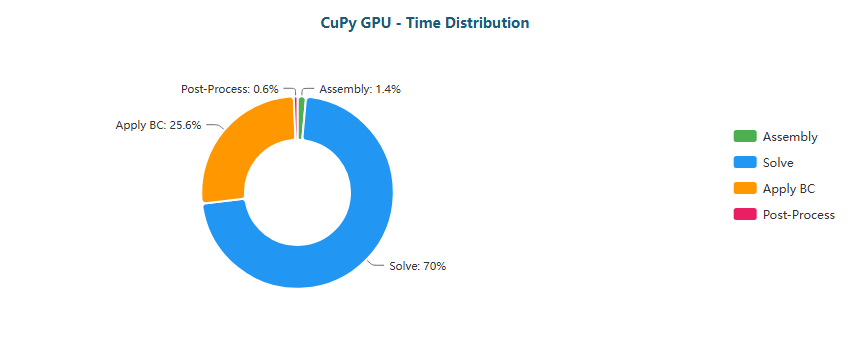
#### S-Bend (XS) - 387 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (68%) | Post-Proc (19%) |
| CPU Threaded | Assembly (50%) | Post-Proc (21%) |
| CPU Multiprocess | Post-Proc (62%) | Assembly (38%) |
| Numba CPU | Solve (55%) | BC (18%) |
| Numba CUDA | Solve (88%) | Assembly (6%) |
| CuPy GPU | Solve (70%) | BC (26%) |

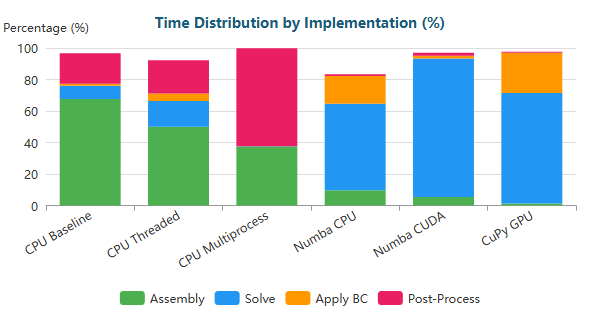
**Time Distribution:**



Pie



Pie



Stacked Bar

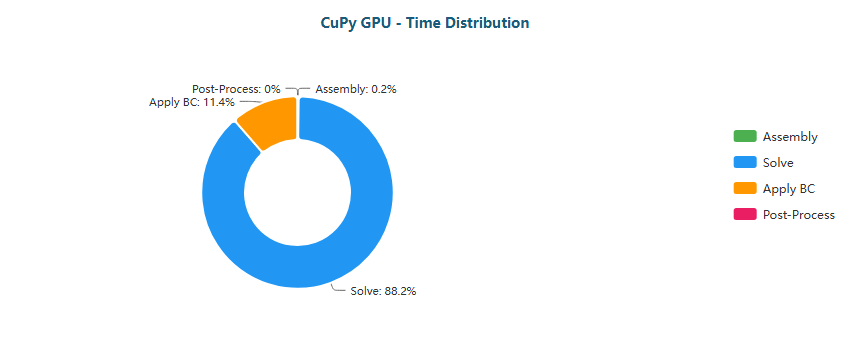
#### S-Bend (M) - 196,078 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (48%) | Assembly (41%) |
| CPU Threaded | Solve (66%) | Assembly (23%) |
| CPU Multiprocess | Solve (49%) | Assembly (21%) |
| Numba CPU | Solve (97%) | BC (2%) |
| Numba CUDA | Solve (52%) | Assembly (30%) |
| CuPy GPU | Solve (88%) | BC (11%) |

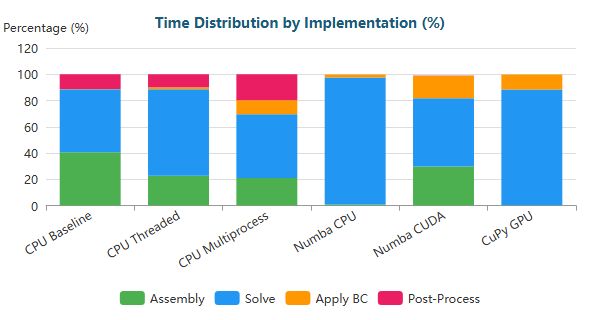
**Time Distribution:**



Pie



Pie



Stacked Bar

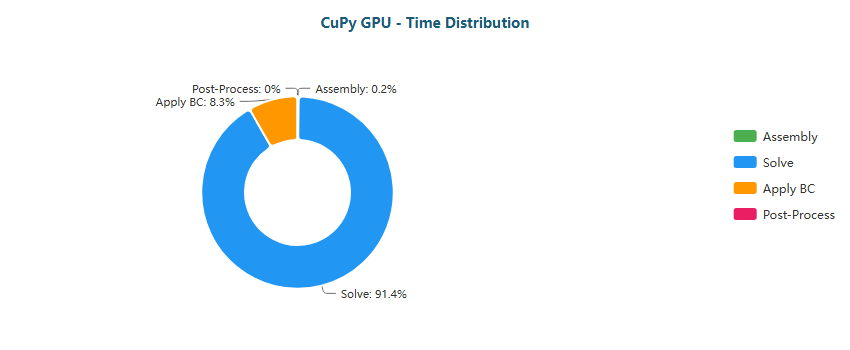
#### S-Bend (L) - 765,441 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (79%) | Assembly (16%) |
| CPU Threaded | Solve (81%) | Assembly (13%) |
| CPU Multiprocess | Solve (77%) | BC (14%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (56%) | Assembly (27%) |
| CuPy GPU | Solve (91%) | BC (8%) |

**Time Distribution:**



Pie



Pie

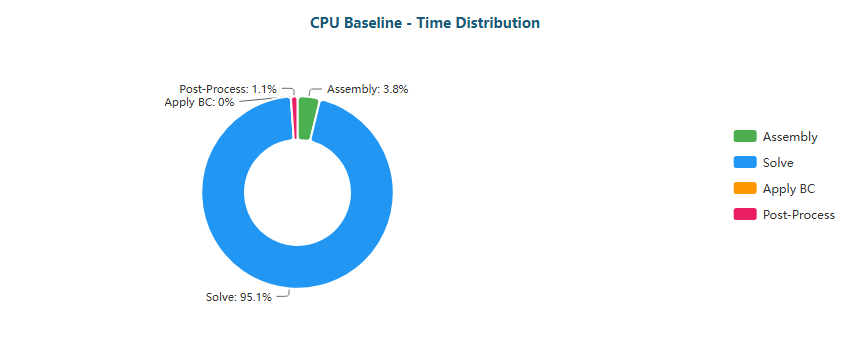


Stacked Bar

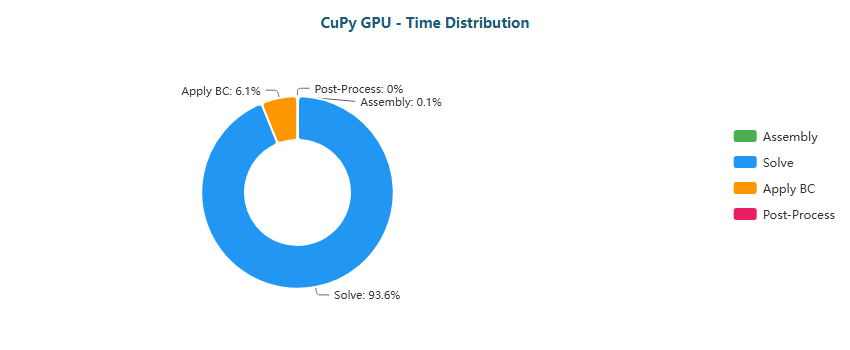
#### S-Bend (XL) - 1,286,039 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (95%) | Assembly (4%) |
| CPU Threaded | Solve (86%) | Assembly (9%) |
| CPU Multiprocess | Solve (83%) | BC (13%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (63%) | Assembly (23%) |
| CuPy GPU | Solve (94%) | BC (6%) |

**Time Distribution:**



Pie



Pie

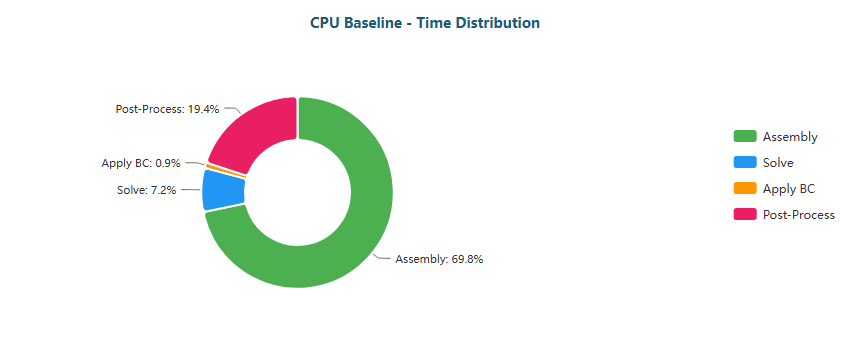


Stacked Bar

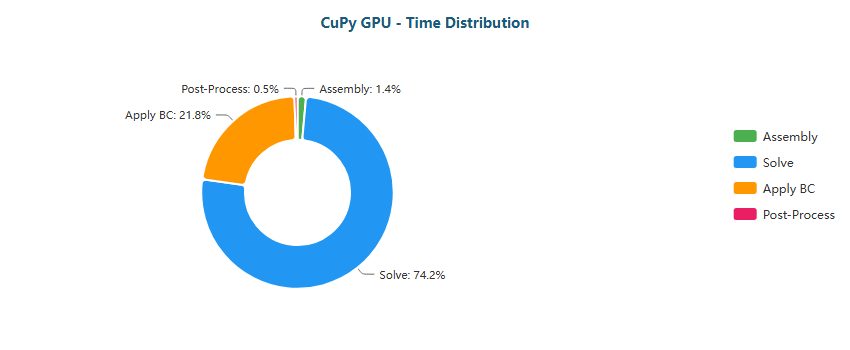
#### T-Junction (XS) - 393 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (70%) | Post-Proc (19%) |
| CPU Threaded | Assembly (54%) | Post-Proc (23%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (54%) | BC (16%) |
| Numba CUDA | Solve (87%) | Assembly (6%) |
| CuPy GPU | Solve (74%) | BC (22%) |

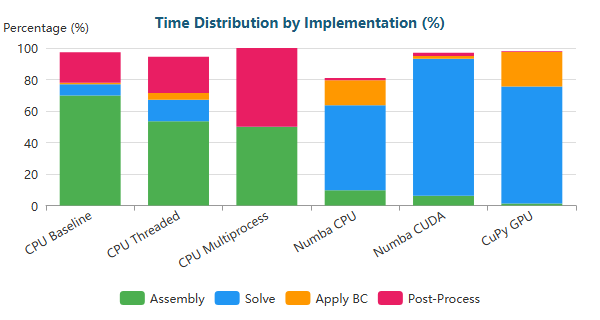
**Time Distribution:**



Pie



Pie

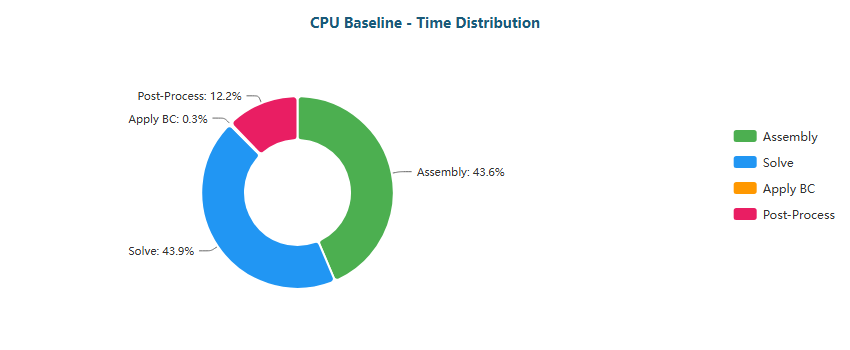


Stacked Bar

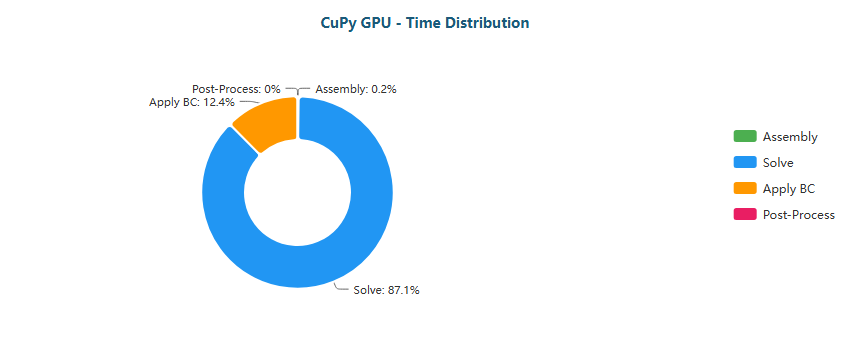
#### T-Junction (M) - 196,420 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (44%) | Assembly (44%) |
| CPU Threaded | Solve (64%) | Assembly (24%) |
| CPU Multiprocess | Solve (46%) | Assembly (22%) |
| Numba CPU | Solve (96%) | BC (3%) |
| Numba CUDA | Solve (51%) | Assembly (31%) |
| CuPy GPU | Solve (87%) | BC (12%) |

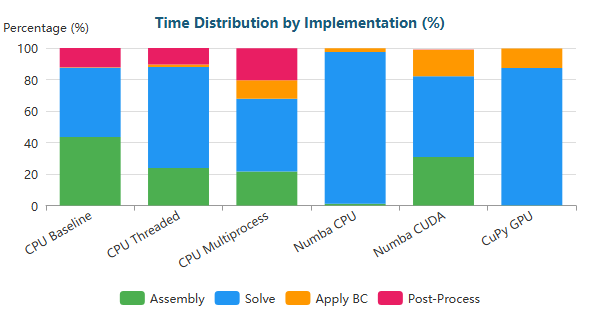
**Time Distribution:**



Pie



Pie

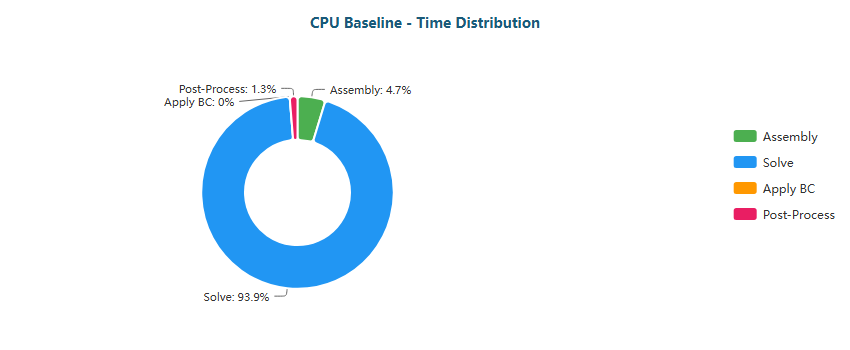


Stacked Bar

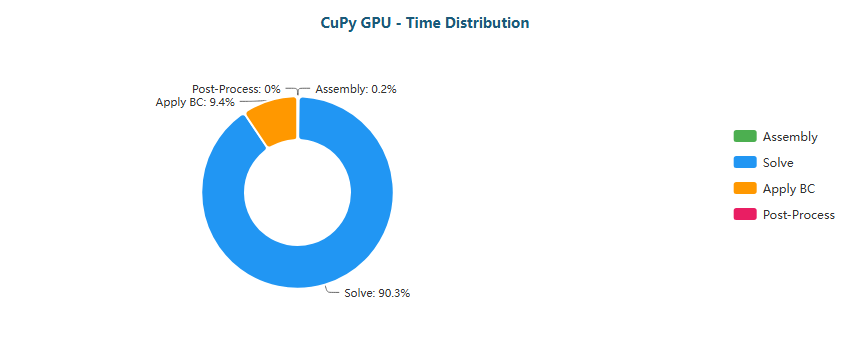
#### T-Junction (L) - 768,898 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (94%) | Assembly (5%) |
| CPU Threaded | Solve (80%) | Assembly (13%) |
| CPU Multiprocess | Solve (76%) | BC (15%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (55%) | Assembly (28%) |
| CuPy GPU | Solve (90%) | BC (9%) |

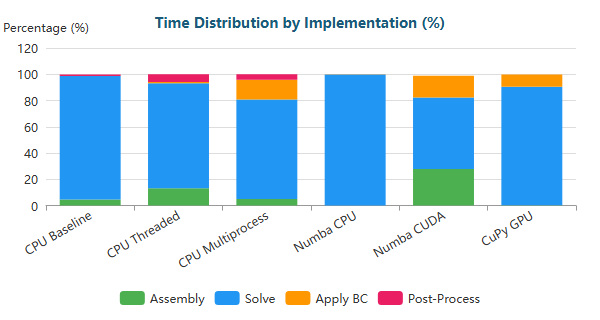
**Time Distribution:**



Pie



Pie

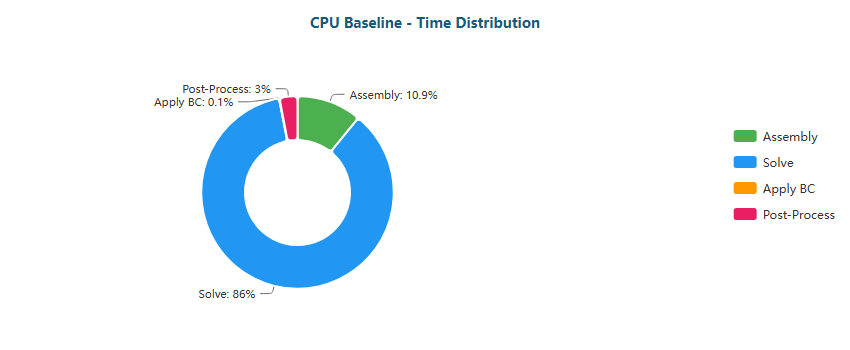


Stacked Bar

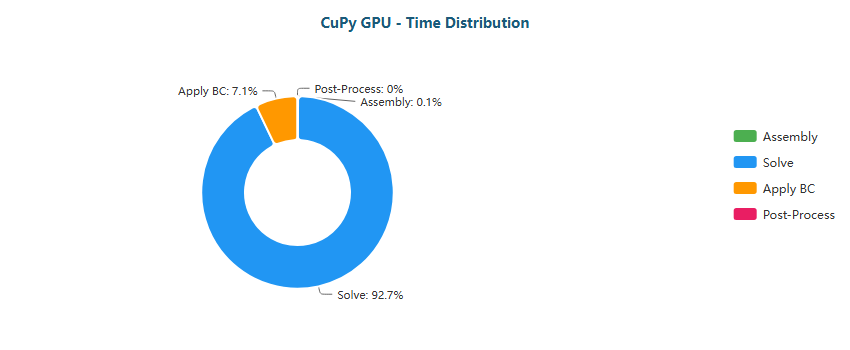
#### T-Junction (XL) - 1,291,289 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (86%) | Assembly (11%) |
| CPU Threaded | Solve (85%) | Assembly (10%) |
| CPU Multiprocess | Solve (81%) | BC (14%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (60%) | Assembly (25%) |
| CuPy GPU | Solve (93%) | BC (7%) |

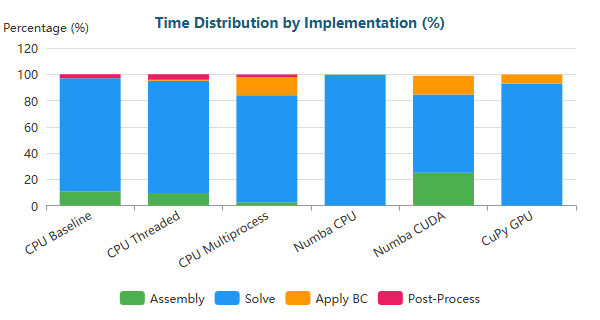
**Time Distribution:**



Pie



Pie

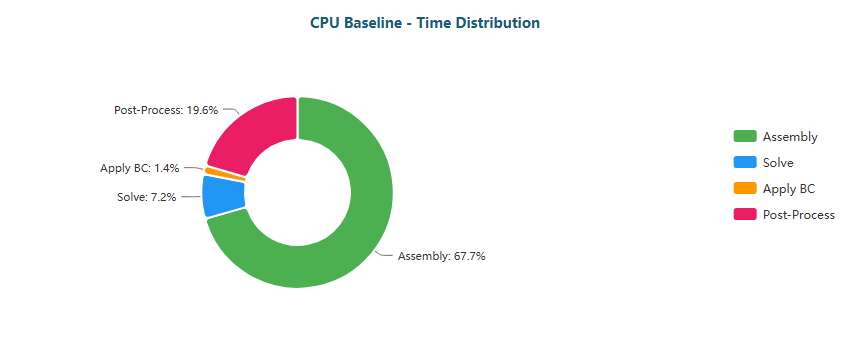


Stacked Bar

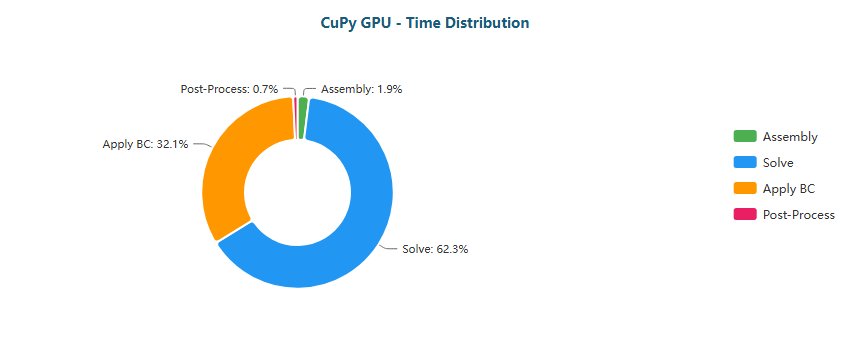
#### Venturi (XS) - 341 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (68%) | Post-Proc (20%) |
| CPU Threaded | Assembly (54%) | Post-Proc (21%) |
| CPU Multiprocess | Post-Proc (62%) | Assembly (38%) |
| Numba CPU | Solve (54%) | BC (17%) |
| Numba CUDA | Solve (85%) | Assembly (7%) |
| CuPy GPU | Solve (62%) | BC (32%) |

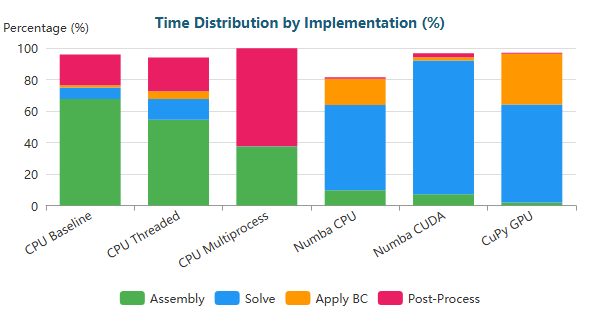
**Time Distribution:**



Pie



Pie

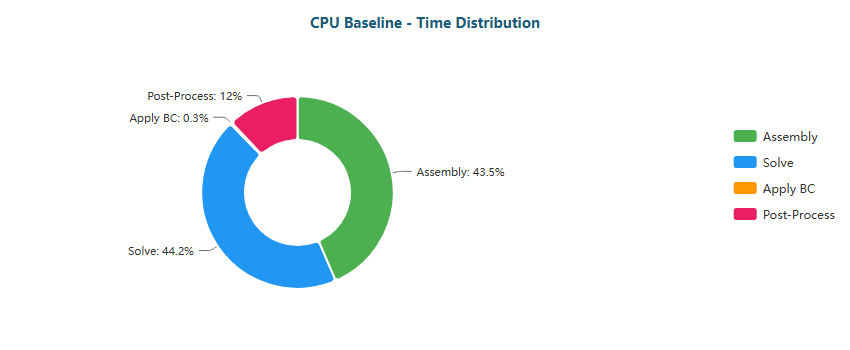


Stacked Bar

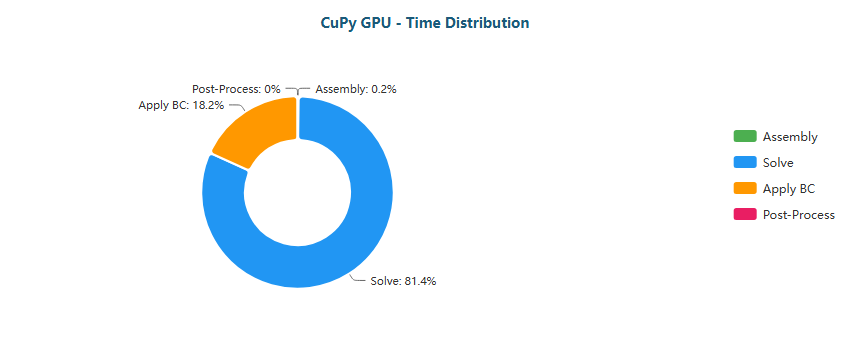
#### Venturi (M) - 194,325 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (44%) | Assembly (44%) |
| CPU Threaded | Solve (63%) | Assembly (25%) |
| CPU Multiprocess | Solve (41%) | Assembly (21%) |
| Numba CPU | Solve (96%) | BC (3%) |
| Numba CUDA | Solve (49%) | Assembly (32%) |
| CuPy GPU | Solve (81%) | BC (18%) |

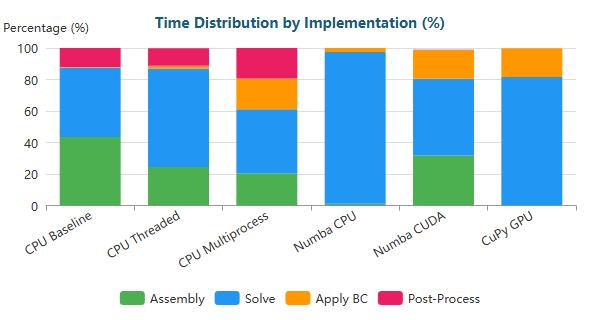
**Time Distribution:**



Pie



Pie

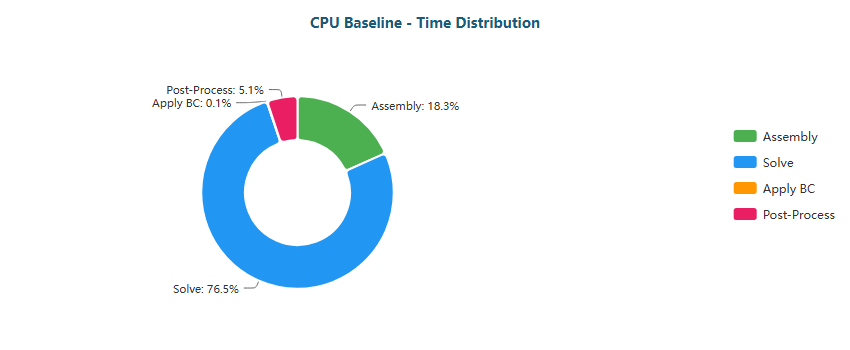


Stacked Bar

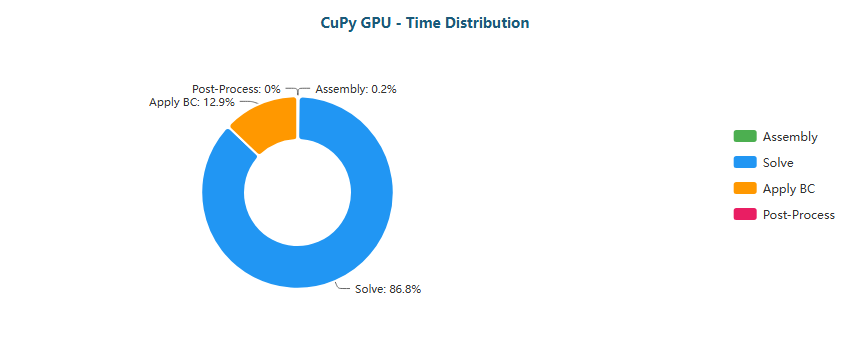
#### Venturi (L) - 763,707 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (77%) | Assembly (18%) |
| CPU Threaded | Solve (78%) | Assembly (15%) |
| CPU Multiprocess | Solve (64%) | BC (27%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (51%) | Assembly (30%) |
| CuPy GPU | Solve (87%) | BC (13%) |

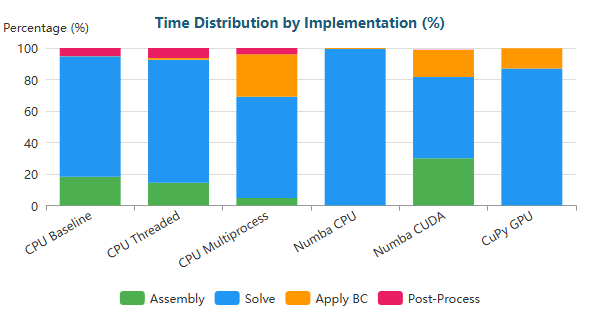
**Time Distribution:**



Pie



Pie

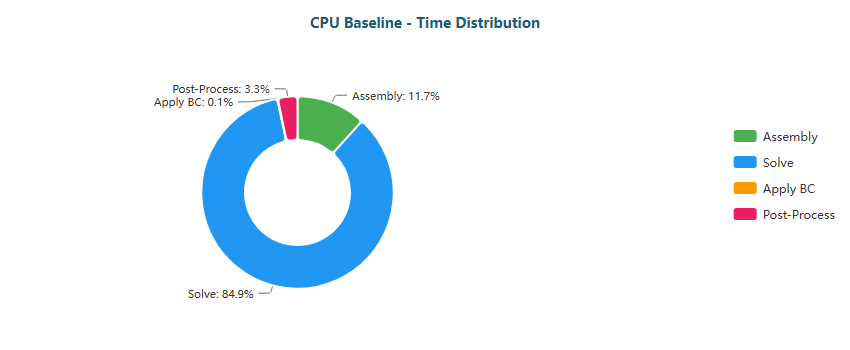


Stacked Bar

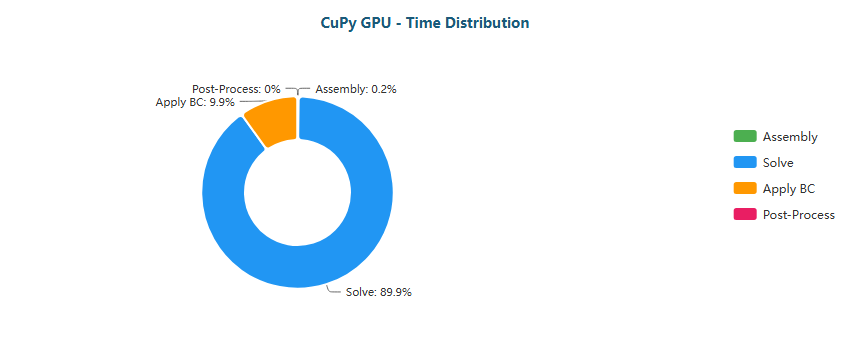
#### Venturi (XL) - 1,284,412 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (85%) | Assembly (12%) |
| CPU Threaded | Solve (84%) | Assembly (11%) |
| CPU Multiprocess | Solve (69%) | BC (26%) |
| Numba CPU | Solve (99%) | BC (0%) |
| Numba CUDA | Solve (57%) | Assembly (26%) |
| CuPy GPU | Solve (90%) | BC (10%) |

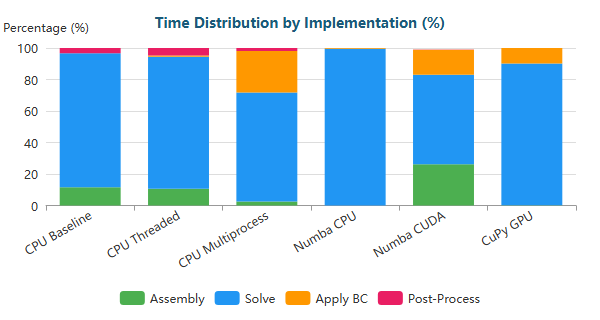
**Time Distribution:**



Pie



Pie



Stacked Bar

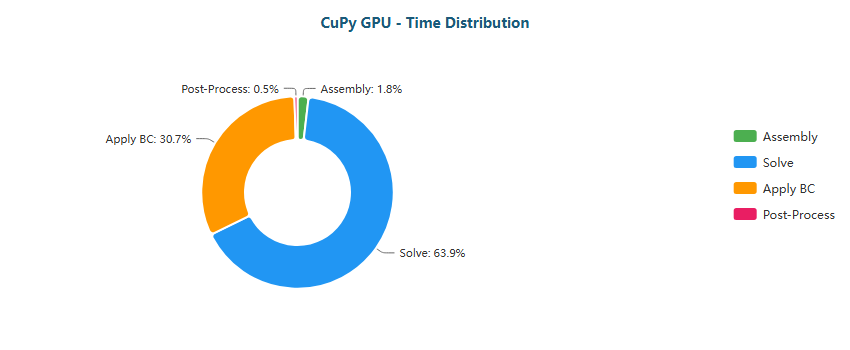
#### Y-Shaped (XS) - 201 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (65%) | Post-Proc (18%) |
| CPU Threaded | Assembly (47%) | Post-Proc (20%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (50%) | BC (14%) |
| Numba CUDA | Solve (87%) | Assembly (6%) |
| CuPy GPU | Solve (64%) | BC (31%) |

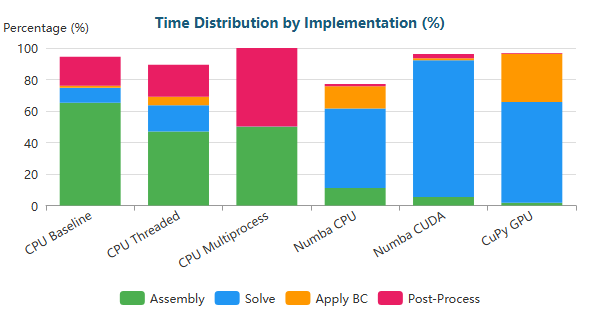
**Time Distribution:**



Pie



Pie

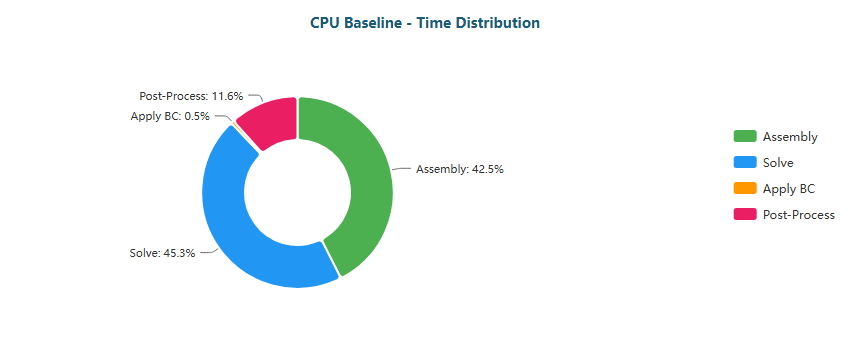


Stacked Bar

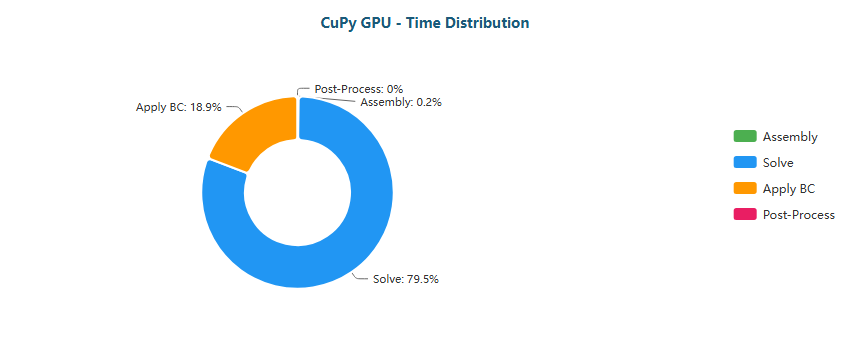
#### Y-Shaped (M) - 195,853 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (45%) | Assembly (43%) |
| CPU Threaded | Solve (65%) | Assembly (23%) |
| CPU Multiprocess | Solve (42%) | Assembly (24%) |
| Numba CPU | Solve (95%) | BC (4%) |
| Numba CUDA | Solve (48%) | Assembly (28%) |
| CuPy GPU | Solve (80%) | BC (19%) |

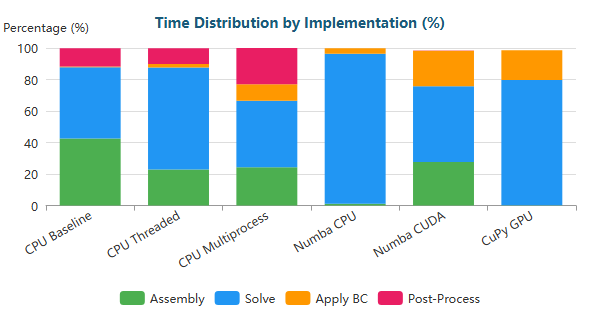
**Time Distribution:**



Pie



Pie

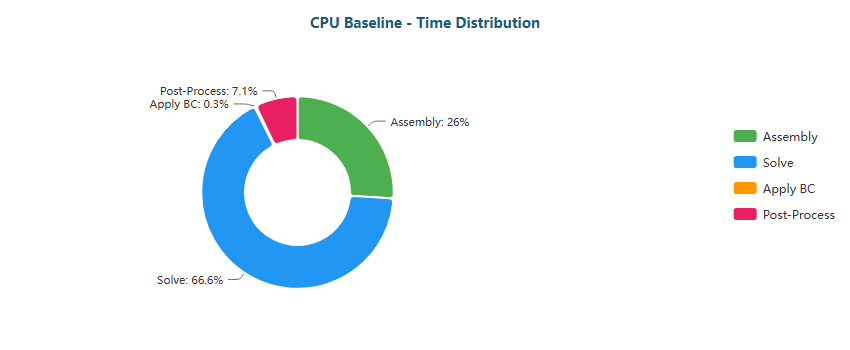


Stacked Bar

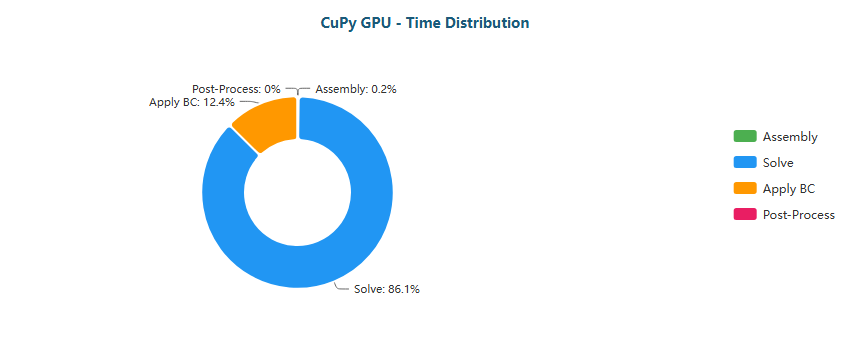
#### Y-Shaped (L) - 772,069 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (67%) | Assembly (26%) |
| CPU Threaded | Solve (81%) | Assembly (12%) |
| CPU Multiprocess | Solve (75%) | BC (15%) |
| Numba CPU | Solve (98%) | BC (2%) |
| Numba CUDA | Solve (51%) | Assembly (26%) |
| CuPy GPU | Solve (86%) | BC (12%) |

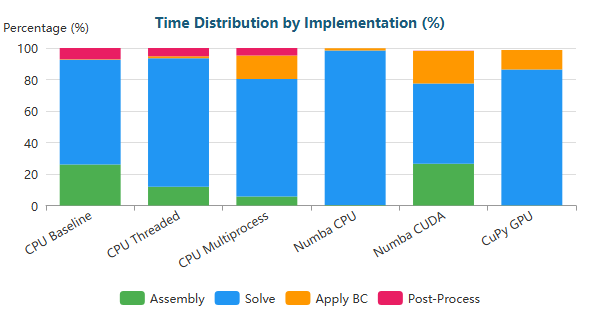
**Time Distribution:**



Pie



Pie

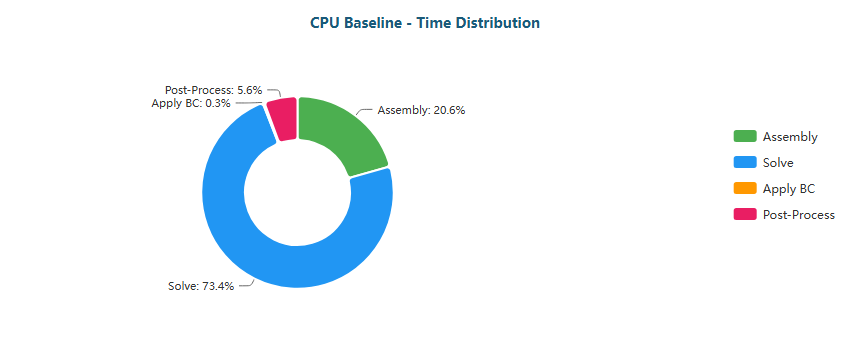


Stacked Bar

#### Y-Shaped (XL) - 1,357,953 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (73%) | Assembly (21%) |
| CPU Threaded | Solve (86%) | Assembly (9%) |
| CPU Multiprocess | Solve (81%) | BC (14%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (57%) | Assembly (23%) |
| CuPy GPU | Solve (90%) | BC (9%) |

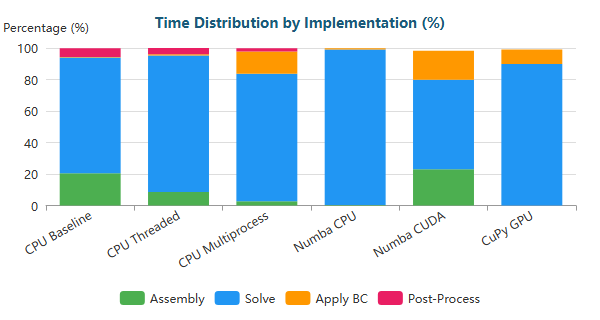
**Time Distribution:**



Pie



Pie



Stacked Bar

### Why Each Optimization Helps

| Transition | Reason |
| --- | --- |
| Baseline → Threaded | Limited by Python GIL; threads only help for I/O |
| Threaded → Multiprocess | Bypasses GIL via separate processes; IPC overhead limits gains |
| Multiprocess → Numba CPU | JIT compilation eliminates interpreter overhead; true parallel loops |
| Numba CPU → Numba CUDA | GPU parallelism: thousands of threads vs dozens of CPU cores |
| Numba CUDA → CuPy GPU | CUDA C kernels more optimized than Numba-generated PTX |

#### 1.3.2.1 Bottleneck Migration Pattern on RTX 4070

Across all five geometries (Backward-Facing Step, Elbow, S-Bend, T-Junction, Venturi, Y-Shaped), the RTX 4070 shows a stable migration sequence:

* **CPU Baseline → CPU Threaded**
  + **XS meshes:** Assembly remains dominant (≈65–70%), Post-Processing stays high (≈18–23%).
  + Threading reduces wall-time but **does not change the dominant stage**, indicating interpreter-level constraints and limited benefit beyond moderate overlap.
* **CPU Multiprocess**
  + A distinctive RTX 4070 signature is the **Post-Processing explosion in XS**:
    - e.g., Backward-Facing Step XS: Post-Proc ≈62% (primary), Assembly ≈38% (secondary).
    - Similar patterns appear in S-Bend XS and Venturi XS, where Post-Proc becomes ≈62%.
  + Interpretation: for tiny meshes, multiprocess introduces enough overhead (IPC + orchestration) that **post-processing becomes a dominant tax**, effectively limiting scaling.
* **Numba CPU (JIT)**
  + The bottleneck systematically transitions to **Solve**, even at moderate scale:
    - M/L/XL: Solve ≈95–100% in almost all cases.
  + For XS, Solve becomes primary (≈48–55%), but **BC becomes visible** (≈14–19%), meaning the pipeline is no longer dominated by assembly once Python overhead is removed.
* **Numba CUDA**
  + At XS, Solve becomes overwhelming (≈85–88%), reflecting that GPU execution makes assembly relatively small.
  + At M/L/XL, Numba CUDA shows a clear **Solve–Assembly split**:
    - Solve ≈48–63%
    - Assembly ≈23–33%
  + Interpretation: on RTX 4070, GPU acceleration exposes that **assembly kernels still cost meaningful time**, often due to memory traffic and scatter operations.
* **CuPy GPU**
  + CuPy converges to a solver-dominated profile for M/L/XL:
    - Solve ≈80–94% (primary)
    - BC ≈6–19% (secondary)
  + For XS, BC becomes unusually prominent:
    - Solve ≈62–74%
    - BC ≈22–32%
  + This indicates that on small meshes the RTX 4070 cannot amortize **launch/sync + BC handling**, making BC a structural overhead.

**Core takeaway:** on RTX 4070, once the GPU is used effectively, the pipeline becomes **solver-bound**, and **Apply BC** is the persistent secondary cost—especially at XS.

#### 1.3.2.2 Optimization Implications and Practical Limits on RTX 4070

The RTX 4070 profile suggests very clear optimization boundaries depending on mesh scale:

* **XS meshes: avoid multiprocess, avoid GPU unless required**
  + Multiprocess is consistently punished by Post-Proc dominance (≈50–62%).
  + GPU modes (Numba CUDA / CuPy) are viable, but BC overhead is large (≈22–32% in CuPy).
  + Practical best choice:
    - **XS:** Numba JIT CPU (best overhead-to-work ratio; bottleneck becomes true computation)
* **M meshes: true crossover zone**
  + CPU Baseline often shows Solve and Assembly competing (≈44–48% each), confirming M is where the pipeline becomes “numerically heavy”.
  + Numba CPU collapses everything into Solve (~96–97%).
  + CuPy becomes solver-dominant (~81–88%) with BC still visible (~11–18%).
  + Practical best choice:
    - **M:** CuPy GPU if end-to-end GPU workflow is available; otherwise Numba CPU is already near-solve-limited.
* **L/XL meshes: solver ceiling dominates**
  + Under CuPy, Solve rises consistently:
    - ≈86–94% across geometries, with BC shrinking to ≈6–13%.
  + This indicates a hard ceiling: further acceleration requires **algorithmic solver gains**, not just kernel tuning.
  + Practical best choice:
    - **L/XL:** CuPy GPU (best scaling; smallest assembly overhead)
* **Where additional speedups still exist**
  + **BC minimization/fusion** matters, especially for XS/M on GPU:
    - reduce sync points
    - avoid multiple passes over global memory
    - fold BC enforcement into fewer kernels
  + **Assembly optimization** mainly matters in Numba CUDA at M/L/XL (Assembly ≈23–33%):
    - fuse operations, reduce scatter traffic, improve memory coalescing
  + **Solver improvements dominate the ceiling** in CuPy:
    - better preconditioning, lower iteration counts, solver choice aligned with matrix structure

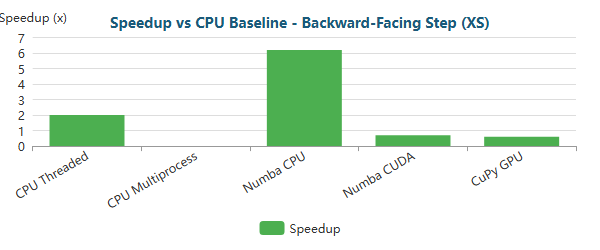
In short, the RTX 4070 behaves like a “solver-dominated GPU” at scale, but exposes stronger **overhead sensitivity** at XS and a sharper **multiprocess penalty** through Post-Processing dominance.

### 1.4 RTX 5060 Ti Performance

Key results from performance benchmarks comparing FEM solver implementations.

**Backward-Facing Step (XS)** (287 nodes)

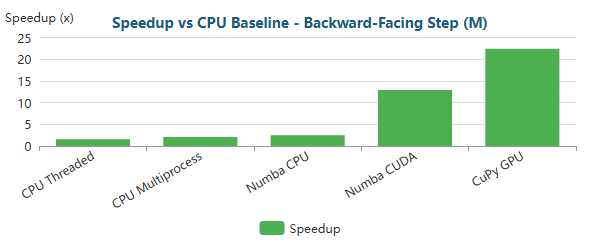
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 33ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 17ms ± 0ms | 2.0x | 3 |
| CPU Multiprocess | 1.57s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 6.2x | 3 |
| Numba CUDA | 49ms ± 3ms | 0.7x | 3 |
| CuPy GPU | 59ms ± 0ms | 0.6x | 3 |



Bar

**Backward-Facing Step (M)** (195,362 nodes)

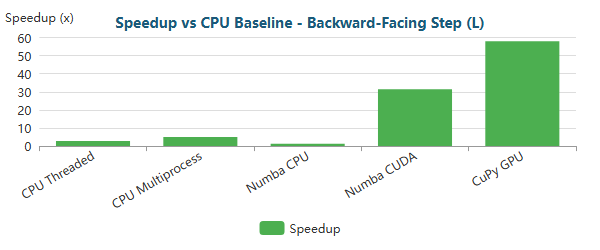
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 36.78s ± 0.03s | 1.0x | 3 |
| CPU Threaded | 23.55s ± 0.04s | 1.6x | 3 |
| CPU Multiprocess | 17.41s ± 0.35s | 2.1x | 3 |
| Numba CPU | 14.68s ± 0.05s | 2.5x | 3 |
| Numba CUDA | 2.85s ± 0.07s | 12.9x | 3 |
| CuPy GPU | 1.64s ± 0.01s | 22.4x | 3 |



Bar

**Backward-Facing Step (L)** (766,088 nodes)

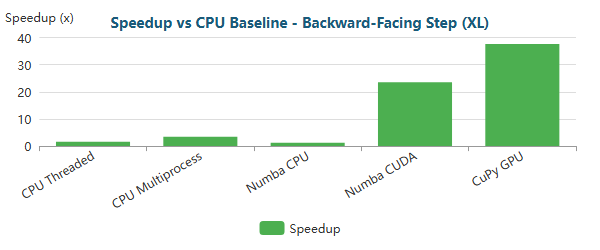
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 32.7s ± 3.8s | 1.0x | 3 |
| CPU Threaded | 2m 18.4s ± 0.3s | 2.8x | 3 |
| CPU Multiprocess | 1m 18.7s ± 1.7s | 5.0x | 3 |
| Numba CPU | 4m 57.4s ± 2.3s | 1.3x | 3 |
| Numba CUDA | 12.51s ± 0.10s | 31.4x | 3 |
| CuPy GPU | 6.78s ± 0.01s | 57.9x | 3 |



Bar

**Backward-Facing Step (XL)** (1,283,215 nodes)

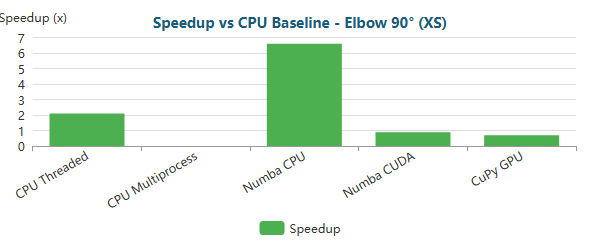
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 8m 57.9s ± 4.5s | 1.0x | 3 |
| CPU Threaded | 5m 37.0s ± 1.9s | 1.6x | 3 |
| CPU Multiprocess | 2m 38.4s ± 2.9s | 3.4x | 3 |
| Numba CPU | 7m 26.5s ± 52.6s | 1.2x | 3 |
| Numba CUDA | 22.88s ± 0.16s | 23.5x | 3 |
| CuPy GPU | 14.31s ± 0.02s | 37.6x | 3 |



Bar

**Elbow 90° (XS)** (411 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 44ms ± 1ms | 1.0x | 3 |
| CPU Threaded | 21ms ± 0ms | 2.1x | 3 |
| CPU Multiprocess | 1.76s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 6.6x | 3 |
| Numba CUDA | 50ms ± 1ms | 0.9x | 3 |
| CuPy GPU | 66ms ± 1ms | 0.7x | 3 |



Bar

**Elbow 90° (M)** (161,984 nodes)

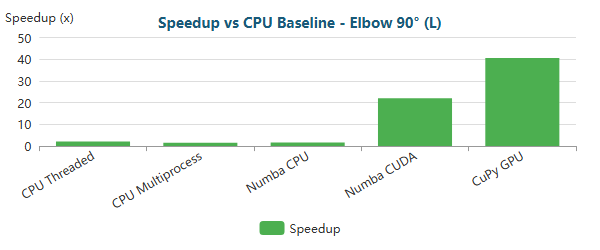
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 31.71s ± 0.09s | 1.0x | 3 |
| CPU Threaded | 17.18s ± 0.08s | 1.8x | 3 |
| CPU Multiprocess | 25.27s ± 0.34s | 1.3x | 3 |
| Numba CPU | 13.79s ± 0.05s | 2.3x | 3 |
| Numba CUDA | 2.30s ± 0.04s | 13.8x | 3 |
| CuPy GPU | 1.30s ± 0.00s | 24.3x | 3 |



Bar

**Elbow 90° (L)** (623,153 nodes)

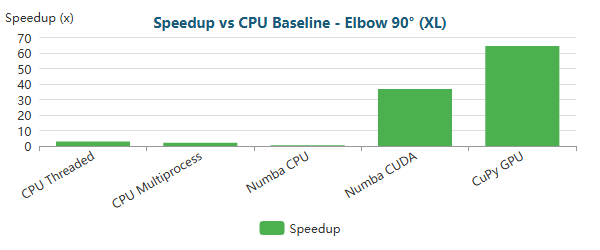
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 3m 11.6s ± 0.5s | 1.0x | 3 |
| CPU Threaded | 1m 33.4s ± 0.1s | 2.1x | 3 |
| CPU Multiprocess | 2m 10.7s ± 3.3s | 1.5x | 3 |
| Numba CPU | 1m 57.7s ± 0.6s | 1.6x | 3 |
| Numba CUDA | 8.71s ± 0.19s | 22.0x | 3 |
| CuPy GPU | 4.72s ± 0.01s | 40.6x | 3 |



Bar

**Elbow 90° (XL)** (1,044,857 nodes)

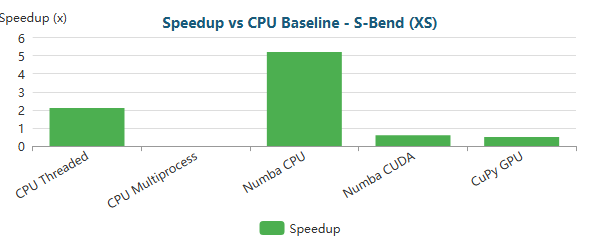
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 9m 54.1s ± 2.5s | 1.0x | 3 |
| CPU Threaded | 3m 19.9s ± 0.2s | 3.0x | 3 |
| CPU Multiprocess | 4m 30.1s ± 5.7s | 2.2x | 3 |
| Numba CPU | 16m 46.7s ± 10.8s | 0.6x | 3 |
| Numba CUDA | 16.16s ± 0.05s | 36.8x | 3 |
| CuPy GPU | 9.21s ± 0.08s | 64.5x | 3 |



Bar

**S-Bend (XS)** (387 nodes)

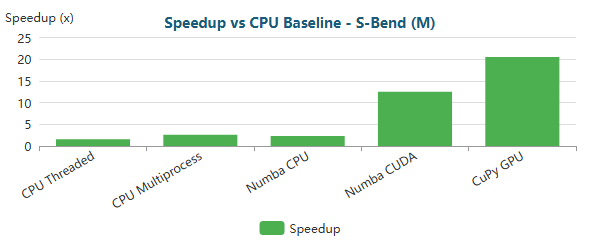
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 37ms ± 2ms | 1.0x | 3 |
| CPU Threaded | 17ms ± 0ms | 2.1x | 3 |
| CPU Multiprocess | 1.90s ± 0.01s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 5.2x | 3 |
| Numba CUDA | 57ms ± 3ms | 0.6x | 3 |
| CuPy GPU | 72ms ± 3ms | 0.5x | 3 |



Bar

**S-Bend (M)** (196,078 nodes)

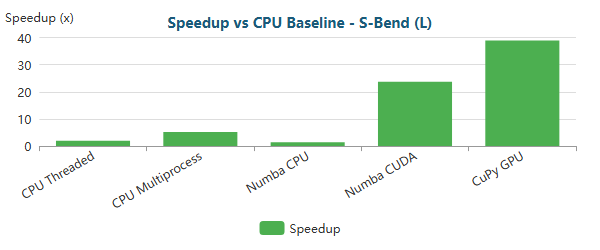
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 39.32s ± 0.08s | 1.0x | 3 |
| CPU Threaded | 24.78s ± 0.04s | 1.6x | 3 |
| CPU Multiprocess | 14.96s ± 0.14s | 2.6x | 3 |
| Numba CPU | 17.27s ± 0.02s | 2.3x | 3 |
| Numba CUDA | 3.14s ± 0.04s | 12.5x | 3 |
| CuPy GPU | 1.92s ± 0.01s | 20.5x | 3 |



Bar

**S-Bend (L)** (765,441 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 5m 9.8s ± 2.1s | 1.0x | 3 |
| CPU Threaded | 2m 31.7s ± 0.3s | 2.0x | 3 |
| CPU Multiprocess | 59.35s ± 0.81s | 5.2x | 3 |
| Numba CPU | 3m 42.3s ± 1.1s | 1.4x | 3 |
| Numba CUDA | 13.09s ± 0.25s | 23.7x | 3 |
| CuPy GPU | 7.97s ± 0.07s | 38.9x | 3 |



Bar

**S-Bend (XL)** (1,286,039 nodes)

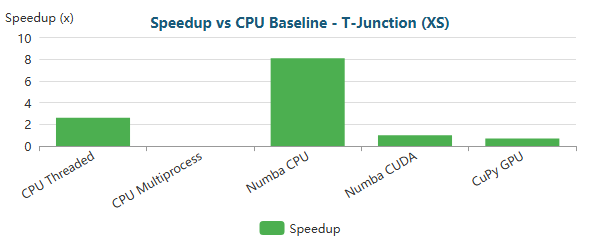
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 51m 34.8s ± 22.2s | 1.0x | 3 |
| CPU Threaded | 6m 28.8s ± 11.1s | 8.0x | 3 |
| CPU Multiprocess | 1m 56.0s ± 0.6s | 26.7x | 3 |
| Numba CPU | 117m 1.0s ± 73.5s | 0.4x | 3 |
| Numba CUDA | 25.90s ± 0.27s | 119.5x | 3 |
| CuPy GPU | 17.16s ± 0.01s | 180.3x | 3 |



Bar

**T-Junction (XS)** (393 nodes)

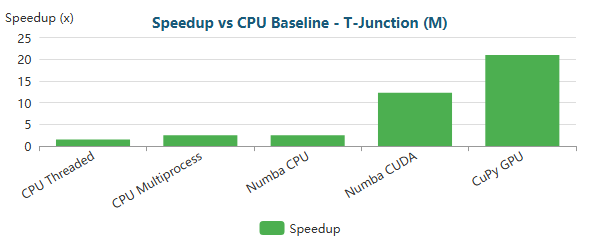
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 56ms ± 9ms | 1.0x | 3 |
| CPU Threaded | 22ms ± 0ms | 2.6x | 3 |
| CPU Multiprocess | 1.85s ± 0.03s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 8.1x | 3 |
| Numba CUDA | 59ms ± 1ms | 1.0x | 3 |
| CuPy GPU | 75ms ± 3ms | 0.7x | 3 |



Bar

**T-Junction (M)** (196,420 nodes)

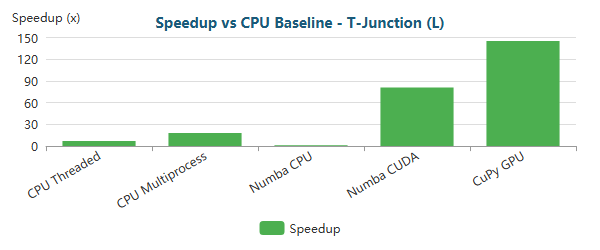
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 37.42s ± 0.03s | 1.0x | 3 |
| CPU Threaded | 24.81s ± 0.27s | 1.5x | 3 |
| CPU Multiprocess | 14.97s ± 0.23s | 2.5x | 3 |
| Numba CPU | 15.04s ± 0.06s | 2.5x | 3 |
| Numba CUDA | 3.03s ± 0.05s | 12.3x | 3 |
| CuPy GPU | 1.78s ± 0.00s | 21.0x | 3 |



Bar

**T-Junction (L)** (768,898 nodes)

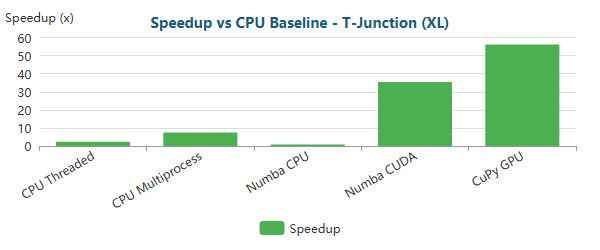
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 17m 31.1s ± 104.0s | 1.0x | 3 |
| CPU Threaded | 2m 25.7s ± 0.3s | 7.2x | 3 |
| CPU Multiprocess | 57.93s ± 0.46s | 18.1x | 3 |
| Numba CPU | 17m 52.8s ± 6.1s | 1.0x | 3 |
| Numba CUDA | 12.97s ± 0.12s | 81.1x | 3 |
| CuPy GPU | 7.24s ± 0.00s | 145.2x | 3 |



Bar

**T-Junction (XL)** (1,291,289 nodes)

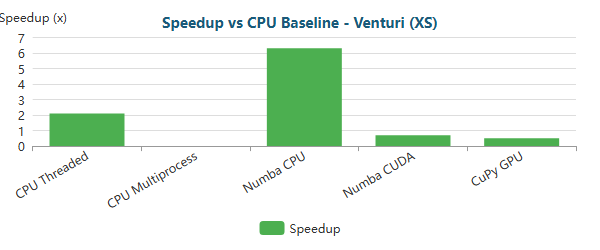
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 14m 10.7s ± 28.2s | 1.0x | 3 |
| CPU Threaded | 5m 56.6s ± 3.5s | 2.4x | 3 |
| CPU Multiprocess | 1m 53.6s ± 2.6s | 7.5x | 3 |
| Numba CPU | 15m 38.3s ± 62.4s | 0.9x | 3 |
| Numba CUDA | 24.03s ± 0.17s | 35.4x | 3 |
| CuPy GPU | 15.14s ± 0.02s | 56.2x | 3 |



Bar

**Venturi (XS)** (341 nodes)

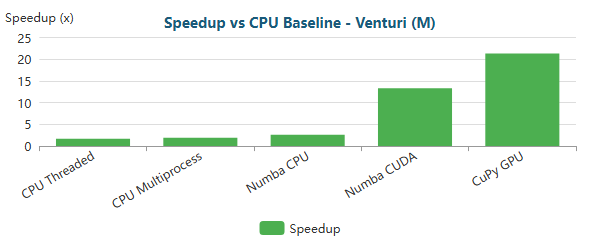
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 36ms ± 2ms | 1.0x | 3 |
| CPU Threaded | 17ms ± 0ms | 2.1x | 3 |
| CPU Multiprocess | 1.54s ± 0.00s | 0.0x | 3 |
| Numba CPU | <0.01s ± 0ms | 6.3x | 3 |
| Numba CUDA | 49ms ± 4ms | 0.7x | 3 |
| CuPy GPU | 68ms ± 3ms | 0.5x | 3 |



Bar

**Venturi (M)** (194,325 nodes)

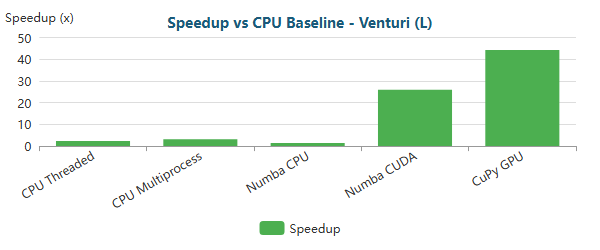
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 38.68s ± 0.18s | 1.0x | 3 |
| CPU Threaded | 22.73s ± 0.02s | 1.7x | 3 |
| CPU Multiprocess | 20.07s ± 0.05s | 1.9x | 3 |
| Numba CPU | 15.01s ± 0.01s | 2.6x | 3 |
| Numba CUDA | 2.90s ± 0.02s | 13.3x | 3 |
| CuPy GPU | 1.81s ± 0.01s | 21.3x | 3 |



Bar

**Venturi (L)** (763,707 nodes)

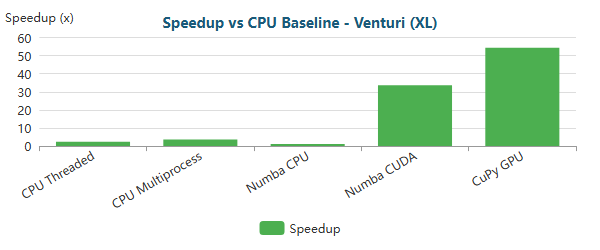
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 5m 3.5s ± 7.6s | 1.0x | 3 |
| CPU Threaded | 2m 11.8s ± 0.2s | 2.3x | 3 |
| CPU Multiprocess | 1m 38.5s ± 1.1s | 3.1x | 3 |
| Numba CPU | 3m 44.8s ± 0.3s | 1.4x | 3 |
| Numba CUDA | 11.72s ± 0.12s | 25.9x | 3 |
| CuPy GPU | 6.85s ± 0.01s | 44.3x | 3 |



Bar

**Venturi (XL)** (1,284,412 nodes)

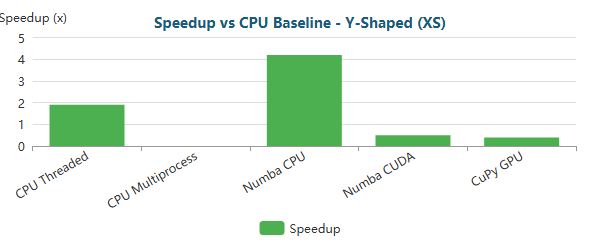
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 12m 35.7s ± 6.1s | 1.0x | 3 |
| CPU Threaded | 5m 17.6s ± 3.6s | 2.4x | 3 |
| CPU Multiprocess | 3m 23.5s ± 2.7s | 3.7x | 3 |
| Numba CPU | 11m 17.9s ± 7.8s | 1.1x | 3 |
| Numba CUDA | 22.46s ± 0.07s | 33.6x | 3 |
| CuPy GPU | 13.91s ± 0.02s | 54.3x | 3 |



Bar

**Y-Shaped (XS)** (201 nodes)

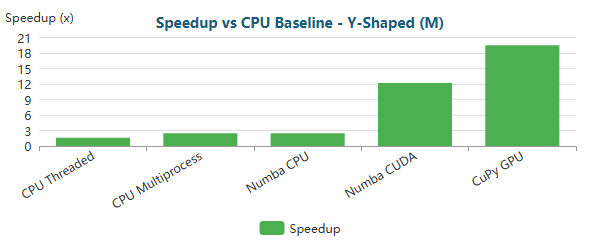
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 22ms ± 0ms | 1.0x | 3 |
| CPU Threaded | 12ms ± 0ms | 1.9x | 3 |
| CPU Multiprocess | 1.23s ± 0.06s | 0.0x | 3 |
| Numba CPU | <0.01s ± 1ms | 4.2x | 3 |
| Numba CUDA | 42ms ± 0ms | 0.5x | 3 |
| CuPy GPU | 57ms ± 3ms | 0.4x | 3 |



Bar

**Y-Shaped (M)** (195,853 nodes)

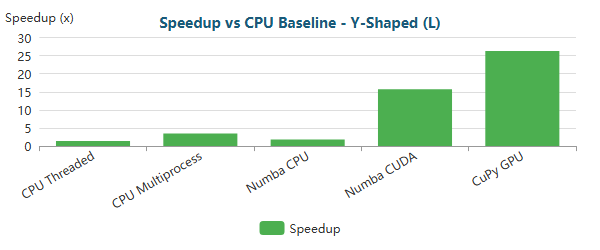
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 31.67s ± 0.64s | 1.0x | 3 |
| CPU Threaded | 19.23s ± 0.04s | 1.6x | 3 |
| CPU Multiprocess | 12.61s ± 0.19s | 2.5x | 3 |
| Numba CPU | 12.80s ± 0.01s | 2.5x | 3 |
| Numba CUDA | 2.60s ± 0.04s | 12.2x | 3 |
| CuPy GPU | 1.63s ± 0.00s | 19.5x | 3 |



Bar

**Y-Shaped (L)** (772,069 nodes)

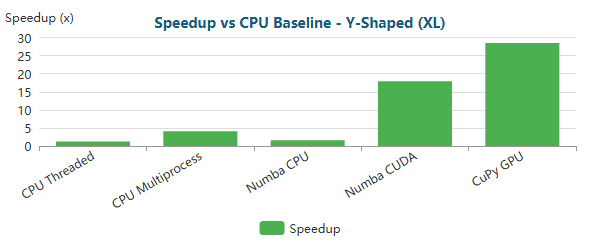
| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 2m 46.4s ± 3.2s | 1.0x | 3 |
| CPU Threaded | 1m 59.8s ± 0.8s | 1.4x | 3 |
| CPU Multiprocess | 47.55s ± 0.92s | 3.5x | 3 |
| Numba CPU | 1m 31.5s ± 0.1s | 1.8x | 3 |
| Numba CUDA | 10.58s ± 0.26s | 15.7x | 3 |
| CuPy GPU | 6.32s ± 0.01s | 26.3x | 3 |



Bar

**Y-Shaped (XL)** (1,357,953 nodes)

| Implementation | Total Time | Speedup vs Baseline | N |
| --- | --- | --- | --- |
| CPU Baseline | 6m 27.7s ± 14.9s | 1.0x | 3 |
| CPU Threaded | 4m 49.3s ± 0.2s | 1.3x | 3 |
| CPU Multiprocess | 1m 35.2s ± 2.3s | 4.1x | 3 |
| Numba CPU | 4m 1.9s ± 1.5s | 1.6x | 3 |
| Numba CUDA | 21.67s ± 0.11s | 17.9x | 3 |
| CuPy GPU | 13.58s ± 0.03s | 28.5x | 3 |



Bar

### 1.4.1 Critical Analysis RTX 5060 Ti

#### 1.4.1.1 Small-Scale Problems (XS meshes)

Across all geometries with **XS meshes (≈200–400 nodes)**, the RTX 5060 Ti exhibits a **strongly overhead-dominated regime**, fully consistent with FEM performance theory:

* **CPU-based implementations dominate** end-to-end execution time.
* **Numba JIT CPU** is the best-performing approach, with speedups typically in the range of **~4.2× to ~8.1×** relative to the CPU baseline (e.g., **6.2×** Backward-Facing Step, **6.6×** Elbow 90°, **5.2×** S-Bend, **8.1×** T-Junction, **6.3×** Venturi).
* **GPU implementations (Numba CUDA and CuPy GPU)** systematically underperform the CPU baseline, with speedups between **~0.4× and ~1.0×**, corresponding to neutral or negative gains.

At this scale, FEM execution is dominated by fixed costs such as kernel launch latency, driver overhead, and synchronization. These costs cannot be amortized with only a few hundred nodes, even on a modern GPU. As a result, **GPU acceleration is clearly ineffective for small FEM problems** on the RTX 5060 Ti.

Additionally, **CPU multiprocessing performs extremely poorly** (often >1s, ≈0× speedup), confirming that process creation and IPC overhead dominate runtime. CPU threading improves performance modestly (~2×), but remains inferior to JIT compilation.

#### 1.4.1.2 Medium-Scale Problems (M meshes)

For **medium-scale meshes (≈160k–200k nodes)**, a consistent **CPU–GPU crossover** emerges across all geometries:

* **GPU acceleration becomes clearly advantageous**.
* **Numba CUDA** achieves speedups of approximately **~12× to ~14×** across all cases.
* **CuPy GPU (RawKernel)** consistently outperforms Numba CUDA, delivering **~19× to ~24×** speedups.
* **CPU-based approaches saturate**, typically limited to **~1.6×–2.6×**, even with threading, multiprocessing, or JIT.

At this scale, arithmetic intensity and parallel workload are sufficient to exploit the GPU effectively. The RTX 5060 Ti shows stable GPU utilization, and CuPy’s lower overhead and better kernel fusion translate into consistently higher performance than Numba CUDA.

#### 1.4.1.3 Large-Scale and Extreme-Scale Problems (L and XL meshes)

For **large (L) and extra-large (XL) meshes (≈700k–1.35M nodes)**, GPU acceleration becomes **essential**:

* **CPU baseline runtimes grow to several minutes**, and in extreme cases exceed **50 minutes** (e.g., S-Bend XL).
* **CPU threading and multiprocessing can provide non-trivial gains** in some geometries (up to **~26×** in S-Bend XL due to solver characteristics), but results are inconsistent and unstable.
* **Numba JIT CPU loses effectiveness**, often matching or underperforming the baseline (≈0.4×–1.6×), reflecting a memory-bound sparse regime.
* **Numba CUDA** achieves speedups of approximately **~16× to ~120×**, depending on geometry and scale.
* **CuPy GPU defines the performance ceiling**, reaching **~26×–180×** speedups at L and XL scales.

At these scales, assembly and post-processing costs are effectively amortized. Performance is dominated by the sparse iterative solver, and speedup curves flatten as execution becomes **memory-bandwidth bound** with irregular access patterns.

#### 1.4.1.4 Comparative Assessment Across Scales

From a practical standpoint, the results support the following execution-model selection for the RTX 5060 Ti:

* **XS meshes:** Numba JIT CPU — minimal overhead and best absolute performance.
* **M meshes:** CuPy GPU (RawKernel) — optimal CPU–GPU crossover and stable gains.
* **L and XL meshes:** CuPy GPU (RawKernel) — maximum scalability and throughput.
* **CPU-only environments:** Numba JIT CPU — best balance of speed and portability.
* **GPU prototyping:** Numba CUDA — faster development with reasonable performance.
* **Production GPU workloads:** CuPy RawKernel — highest and most consistent speedups.

#### 1.4.1.5 Comparative Assessment of Execution Models

From a practical standpoint, the benchmark results support the following conclusions:

| Regime | Best Execution Model | Rationale |
| --- | --- | --- |
| XS meshes | Numba JIT CPU | Minimal overhead, compiled execution |
| M meshes | CuPy GPU (RawKernel) | Maximum throughput, full GPU residency |
| CPU-only environments | Numba JIT CPU | Best balance of speed and memory efficiency |
| GPU prototyping | Numba CUDA | Easier development, acceptable performance |
| Production GPU | CuPy RawKernel | Highest and most consistent speedups |

Overall, the RTX 5060 Ti demonstrates **clear and predictable scalability** once problem size justifies GPU usage. While absolute performance is naturally lower than high-end GPUs, the architectural behavior mirrors that of larger cards: GPU acceleration delivers **order-of-magnitude speedups** for medium to extreme FEM workloads when implemented efficiently. At the same time, the results reinforce several key best practices:

* GPU acceleration must be **selectively applied**, not used indiscriminately.
* Small FEM problems are better served by optimized CPU execution.
* **RawKernel-based GPU implementations provide the highest return on investment**.
* At scale, the **sparse linear solver—not the assembly kernel—becomes the dominant bottleneck**.

These results position the RTX 5060 Ti as a **strong mid-range GPU for large-scale FEM workloads**, while clearly illustrating the limits imposed by problem size, solver structure, and memory behavior rather than raw compute capability alone.

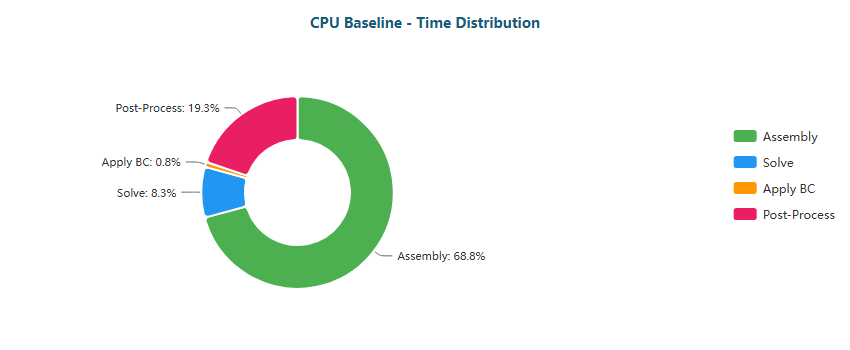
### 1.4.2 RTX 4060Ti Bottleneck Evolution Critical Analysis

As optimizations progress, the computational bottleneck shifts:

#### Backward-Facing Step (XS) - 287 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (69%) | Post-Proc (19%) |
| CPU Threaded | Assembly (50%) | Post-Proc (24%) |
| CPU Multiprocess | Post-Proc (60%) | Assembly (39%) |
| Numba CPU | Solve (59%) | BC (15%) |
| Numba CUDA | Solve (89%) | Assembly (6%) |
| CuPy GPU | Solve (71%) | BC (25%) |

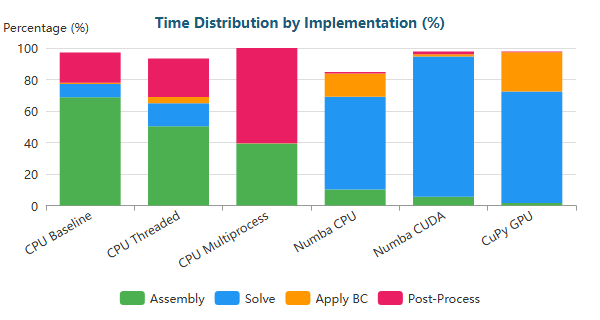
**Time Distribution:**



Pie



Pie

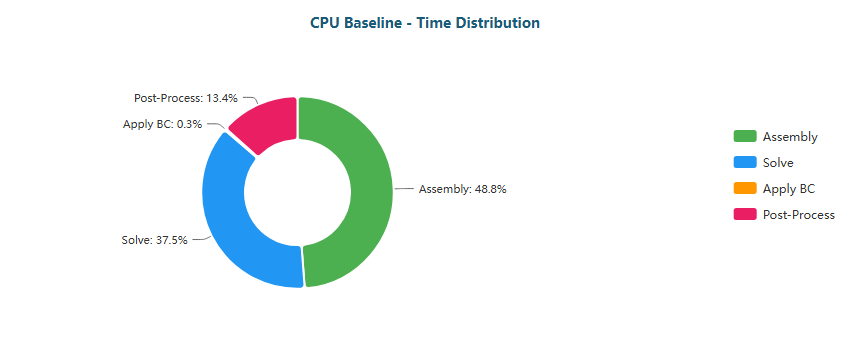


Stacked Bar

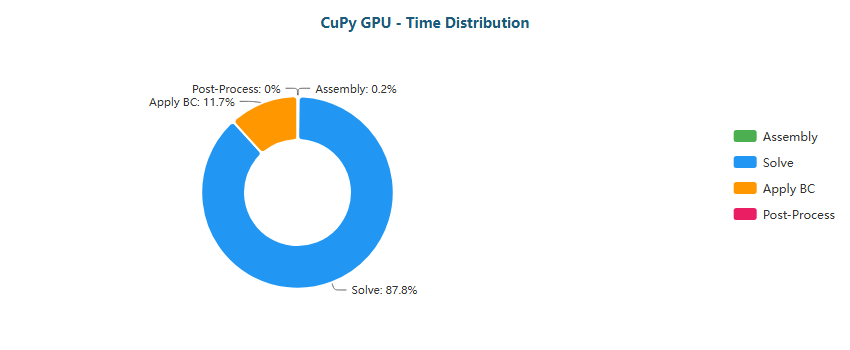
#### Backward-Facing Step (M) - 195,362 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (49%) | Solve (37%) |
| CPU Threaded | Solve (55%) | Assembly (29%) |
| CPU Multiprocess | BC (53%) | Assembly (26%) |
| Numba CPU | Solve (94%) | BC (4%) |
| Numba CUDA | Solve (50%) | Assembly (28%) |
| CuPy GPU | Solve (88%) | BC (12%) |

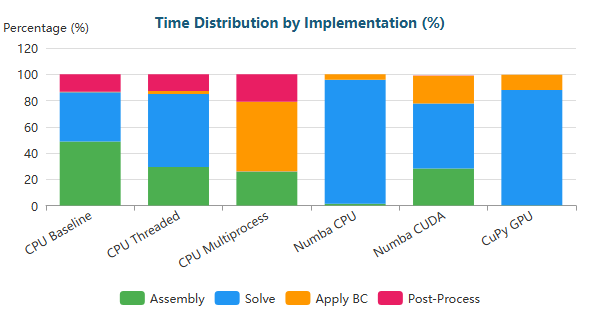
**Time Distribution:**



Pie



Pie

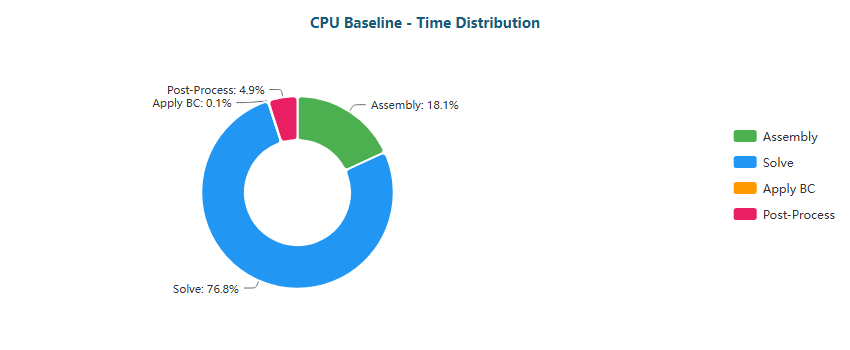


Stacked Bar

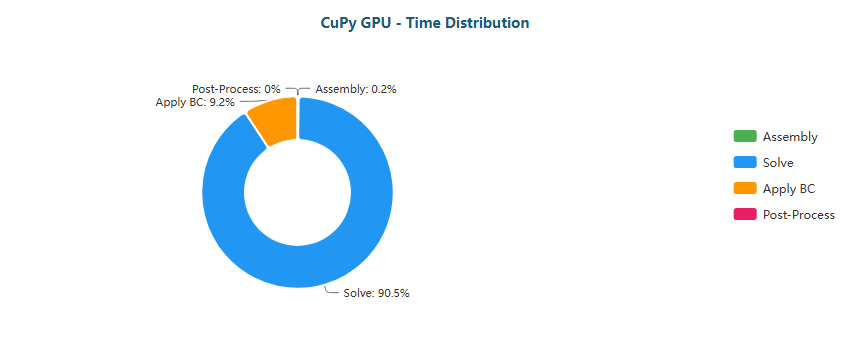
#### Backward-Facing Step (L) - 766,088 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (77%) | Assembly (18%) |
| CPU Threaded | Solve (70%) | Assembly (20%) |
| CPU Multiprocess | BC (84%) | Assembly (10%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (53%) | Assembly (25%) |
| CuPy GPU | Solve (90%) | BC (9%) |

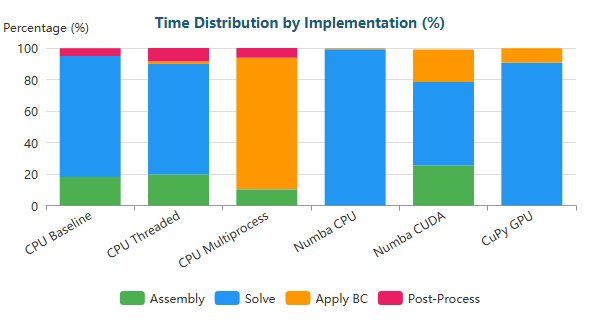
**Time Distribution:**



Pie



Pie

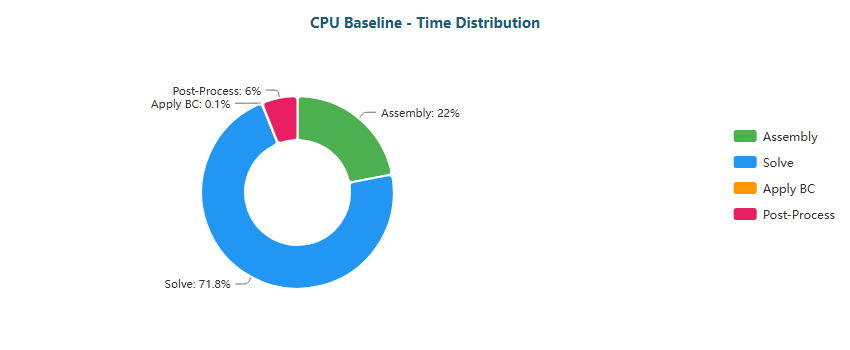


Stacked Bar

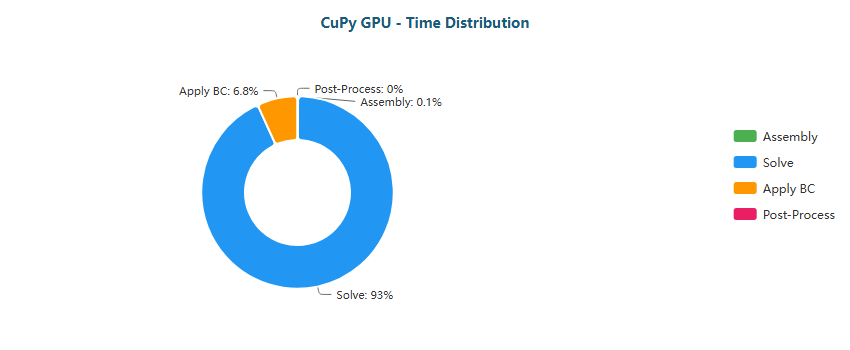
#### Backward-Facing Step (XL) - 1,283,215 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (72%) | Assembly (22%) |
| CPU Threaded | Solve (80%) | Assembly (13%) |
| CPU Multiprocess | BC (89%) | Assembly (7%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (59%) | Assembly (24%) |
| CuPy GPU | Solve (93%) | BC (7%) |

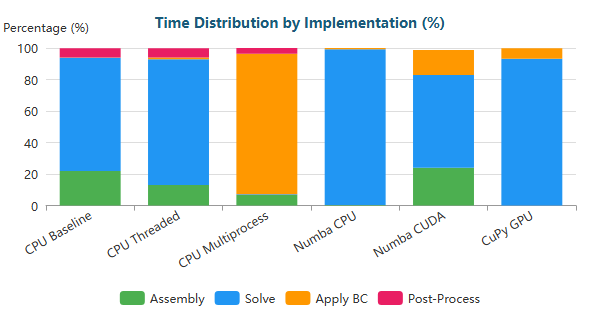
**Time Distribution:**



Pie



Pie

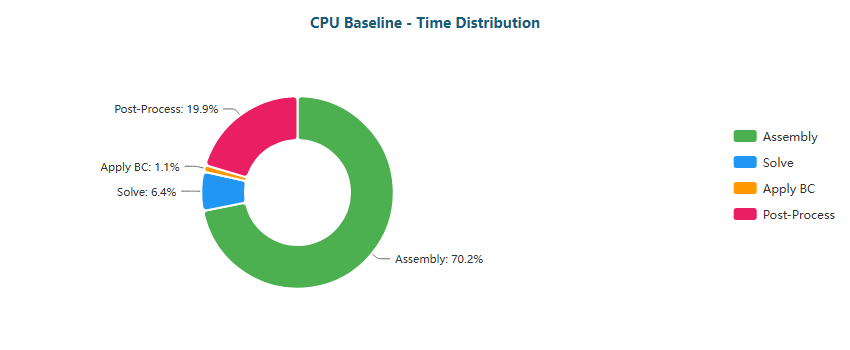


Stacked Bar

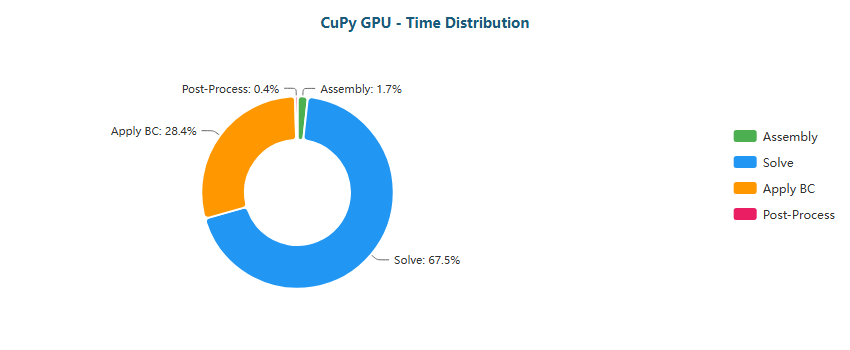
#### Elbow 90° (XS) - 411 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (70%) | Post-Proc (20%) |
| CPU Threaded | Assembly (52%) | Post-Proc (26%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (54%) | BC (19%) |
| Numba CUDA | Solve (88%) | Assembly (6%) |
| CuPy GPU | Solve (67%) | BC (28%) |

**Time Distribution:**



Pie



Pie

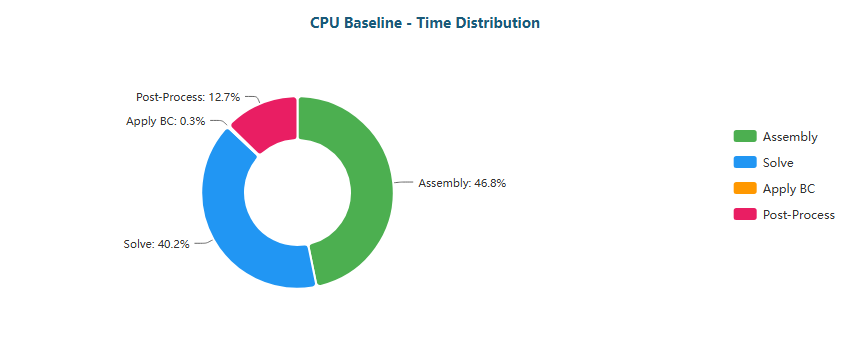


Stacked Bar

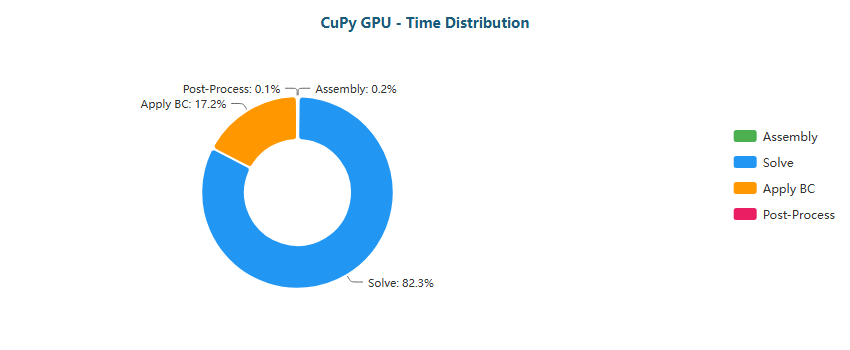
#### Elbow 90° (M) - 161,984 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (47%) | Solve (40%) |
| CPU Threaded | Solve (51%) | Assembly (32%) |
| CPU Multiprocess | BC (69%) | Assembly (17%) |
| Numba CPU | Solve (95%) | BC (4%) |
| Numba CUDA | Solve (47%) | Assembly (29%) |
| CuPy GPU | Solve (82%) | BC (17%) |

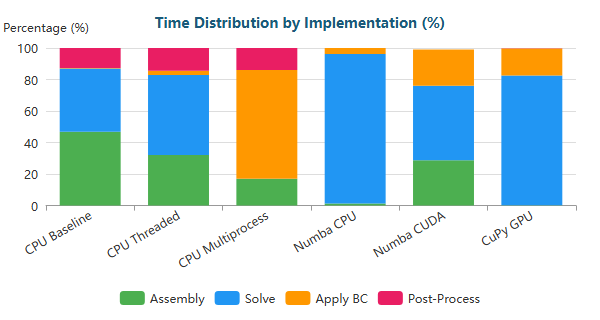
**Time Distribution:**



Pie



Pie

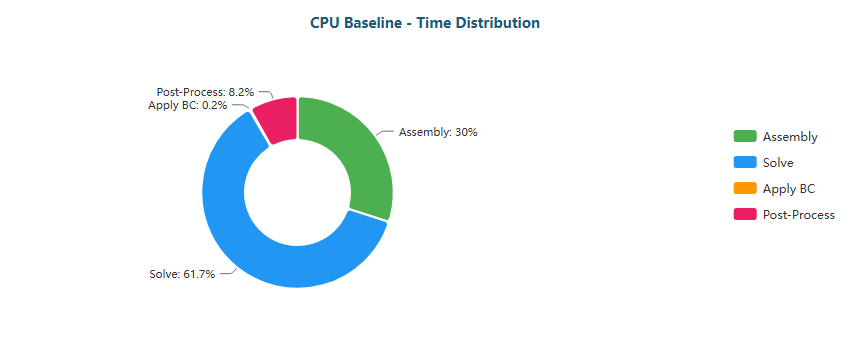


Stacked Bar

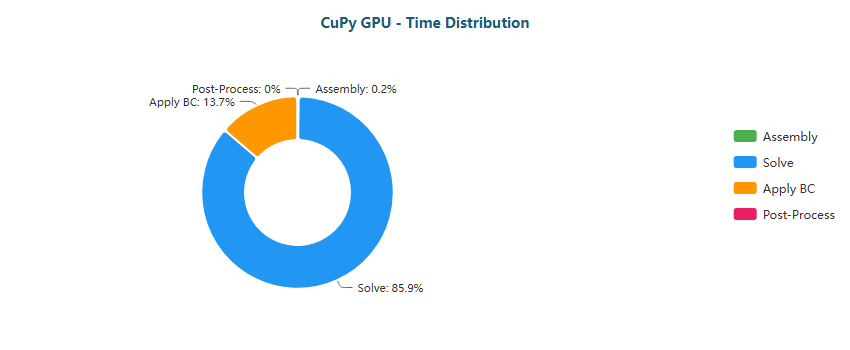
#### Elbow 90° (L) - 623,153 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (62%) | Assembly (30%) |
| CPU Threaded | Solve (65%) | Assembly (23%) |
| CPU Multiprocess | BC (91%) | Assembly (5%) |
| Numba CPU | Solve (98%) | BC (2%) |
| Numba CUDA | Solve (47%) | Assembly (30%) |
| CuPy GPU | Solve (86%) | BC (14%) |

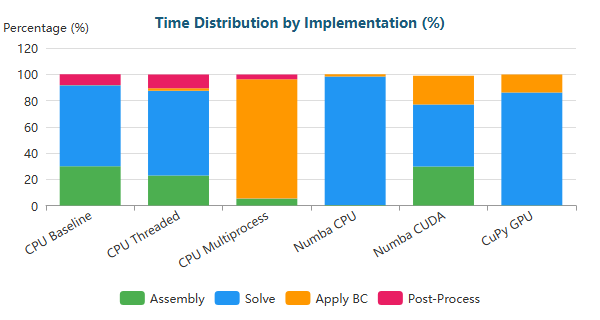
**Time Distribution:**



Pie



Pie

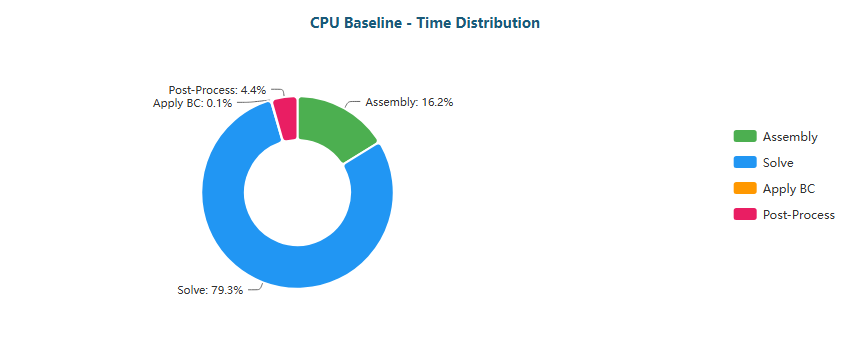


Stacked Bar

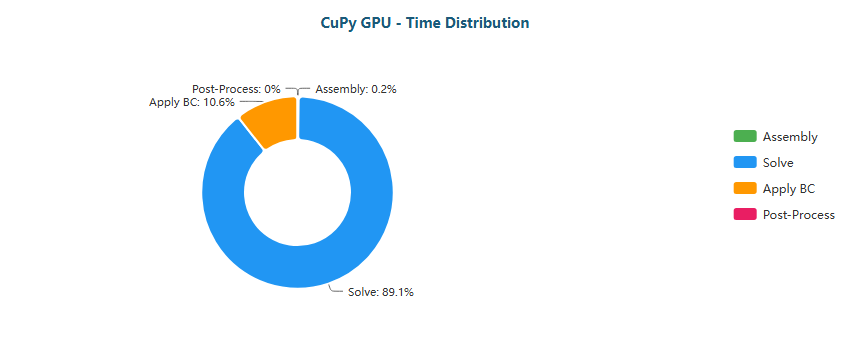
#### Elbow 90° (XL) - 1,044,857 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (79%) | Assembly (16%) |
| CPU Threaded | Solve (72%) | Assembly (18%) |
| CPU Multiprocess | BC (94%) | Assembly (4%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (53%) | Assembly (27%) |
| CuPy GPU | Solve (89%) | BC (11%) |

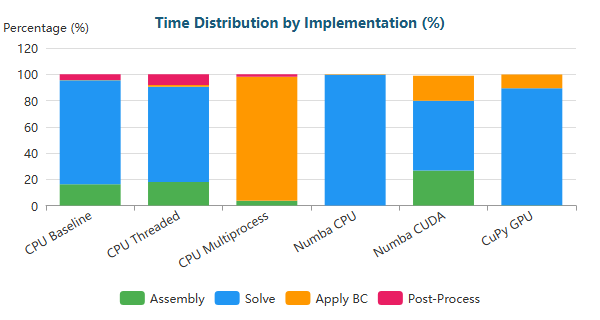
**Time Distribution:**



Pie



Pie



Stacked Bar

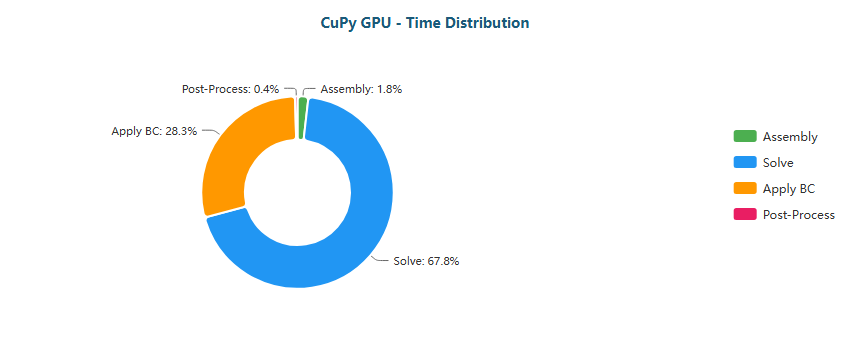
#### S-Bend (XS) - 387 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (65%) | Post-Proc (20%) |
| CPU Threaded | Assembly (50%) | Post-Proc (21%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (61%) | BC (17%) |
| Numba CUDA | Solve (90%) | Assembly (5%) |
| CuPy GPU | Solve (68%) | BC (28%) |

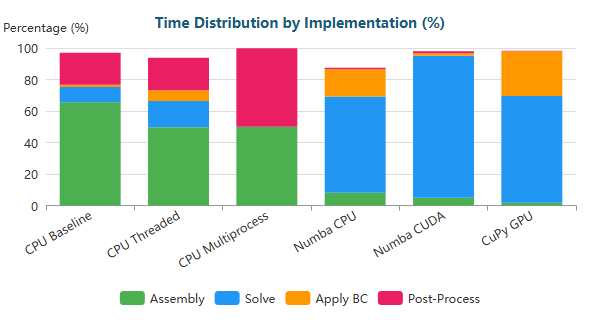
**Time Distribution:**



Pie



Pie

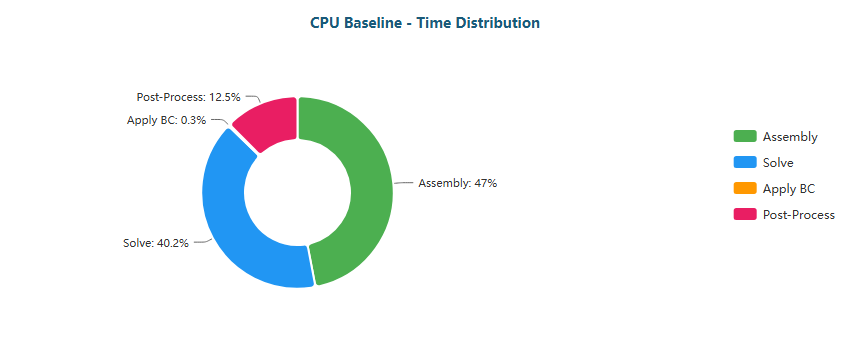


Stacked Bar

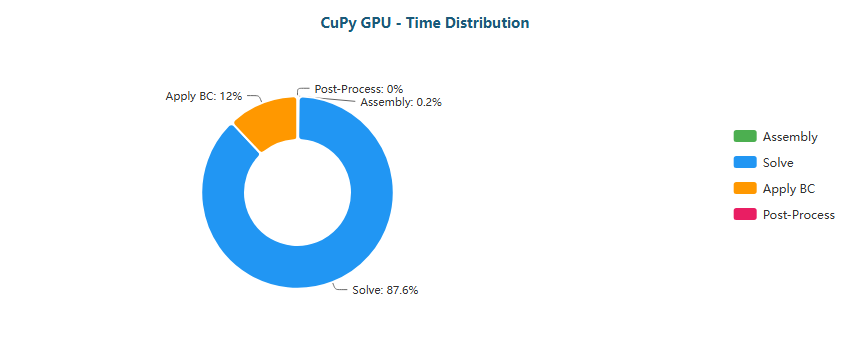
#### S-Bend (M) - 196,078 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (47%) | Solve (40%) |
| CPU Threaded | Solve (61%) | Assembly (25%) |
| CPU Multiprocess | BC (43%) | Assembly (31%) |
| Numba CPU | Solve (95%) | BC (4%) |
| Numba CUDA | Solve (53%) | Assembly (26%) |
| CuPy GPU | Solve (88%) | BC (12%) |

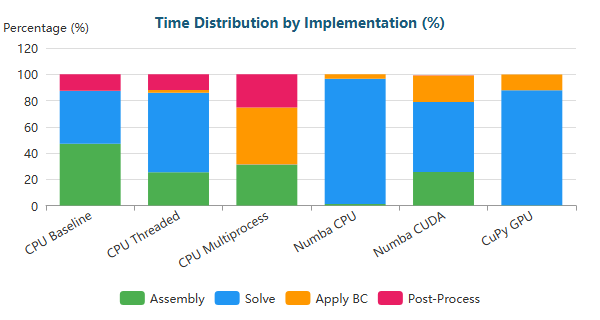
**Time Distribution:**



Pie



Pie

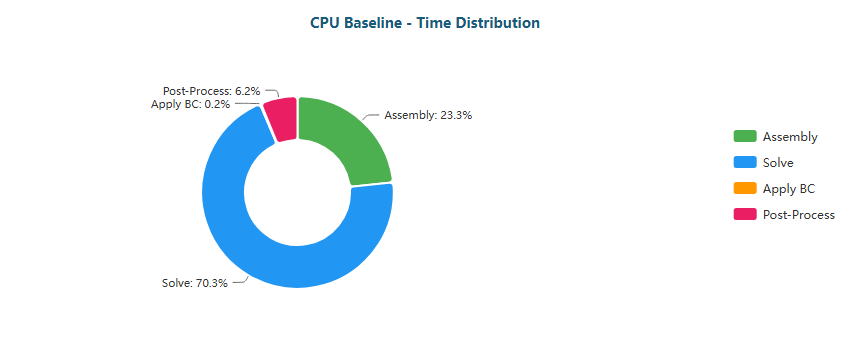


Stacked Bar

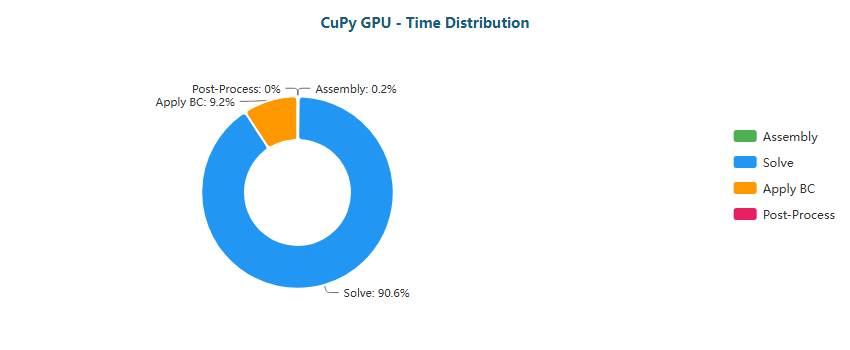
#### S-Bend (L) - 765,441 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (70%) | Assembly (23%) |
| CPU Threaded | Solve (75%) | Assembly (16%) |
| CPU Multiprocess | BC (77%) | Assembly (14%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (56%) | Assembly (24%) |
| CuPy GPU | Solve (91%) | BC (9%) |

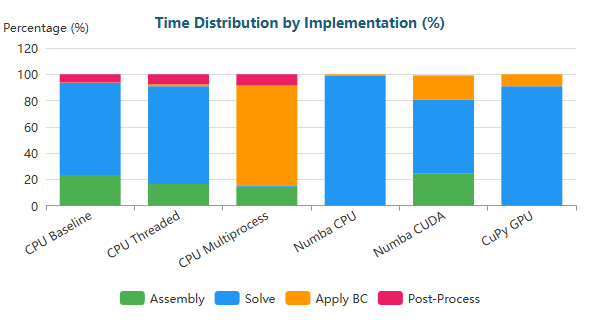
**Time Distribution:**



Pie



Pie

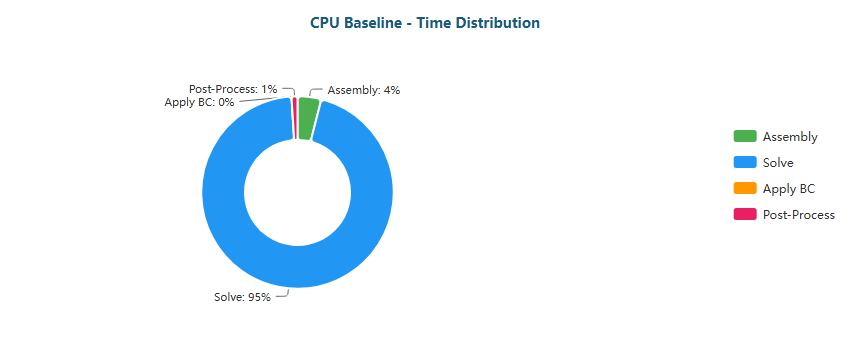


Stacked Bar

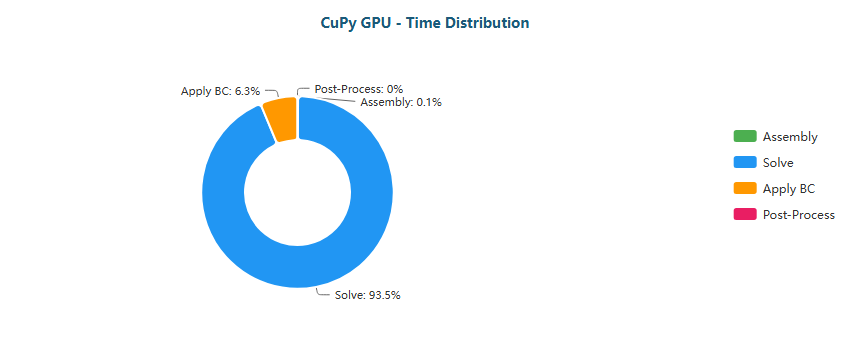
#### S-Bend (XL) - 1,286,039 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (95%) | Assembly (4%) |
| CPU Threaded | Solve (83%) | Assembly (11%) |
| CPU Multiprocess | BC (84%) | Assembly (10%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (64%) | Assembly (21%) |
| CuPy GPU | Solve (94%) | BC (6%) |

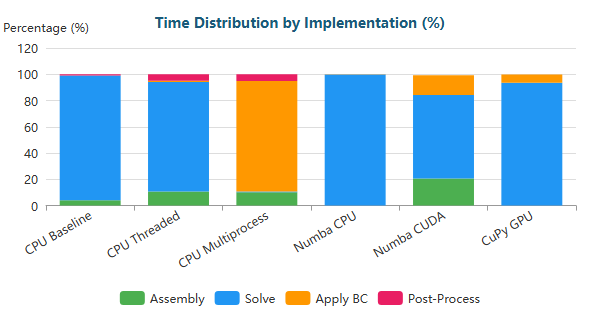
**Time Distribution:**



Pie



Pie

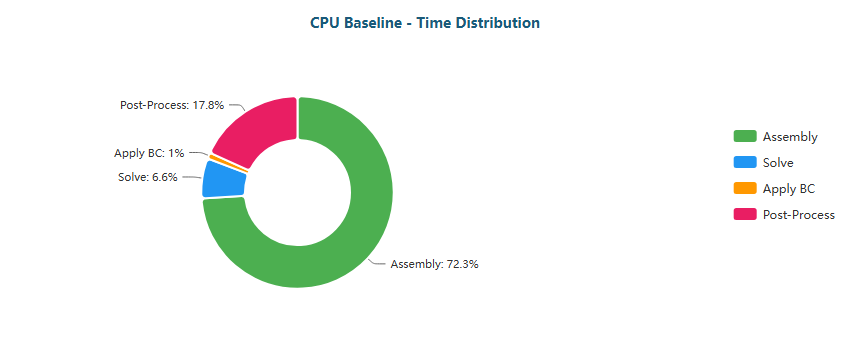


Stacked Bar

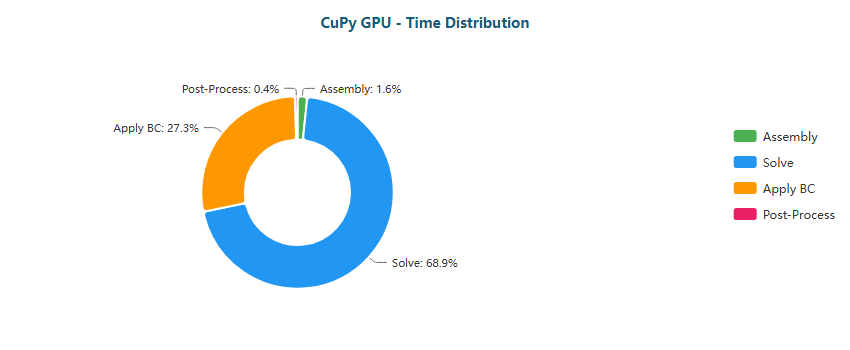
#### T-Junction (XS) - 393 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (72%) | Post-Proc (18%) |
| CPU Threaded | Assembly (51%) | Post-Proc (25%) |
| CPU Multiprocess | Assembly (51%) | Post-Proc (49%) |
| Numba CPU | Solve (58%) | BC (16%) |
| Numba CUDA | Solve (89%) | Assembly (6%) |
| CuPy GPU | Solve (69%) | BC (27%) |

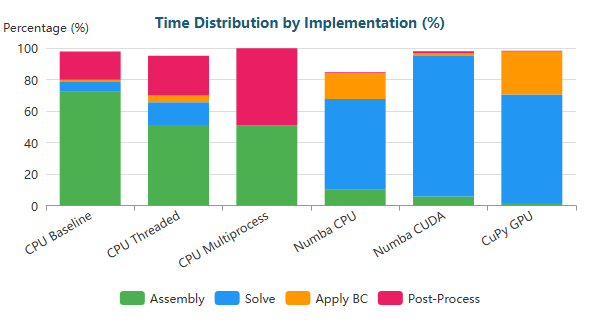
**Time Distribution:**



Pie



Pie

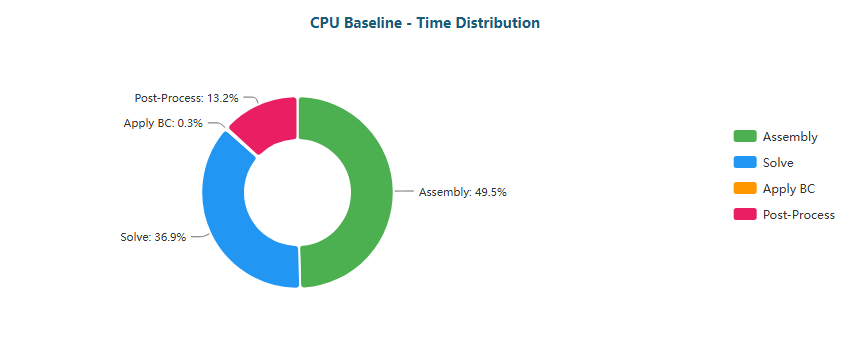


Stacked Bar

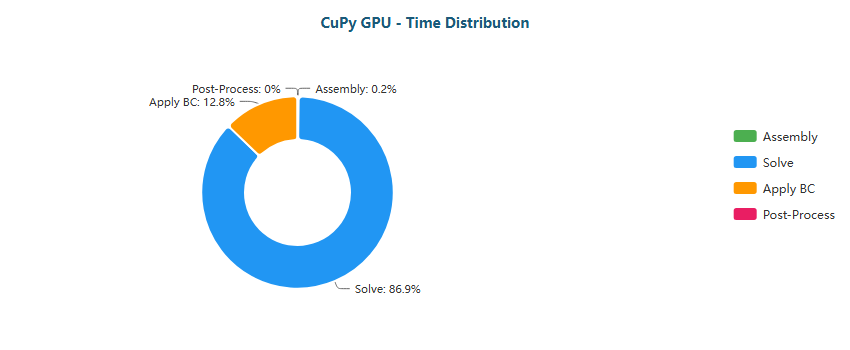
#### T-Junction (M) - 196,420 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (50%) | Solve (37%) |
| CPU Threaded | Solve (58%) | Assembly (27%) |
| CPU Multiprocess | BC (44%) | Assembly (31%) |
| Numba CPU | Solve (95%) | BC (4%) |
| Numba CUDA | Solve (51%) | Assembly (27%) |
| CuPy GPU | Solve (87%) | BC (13%) |

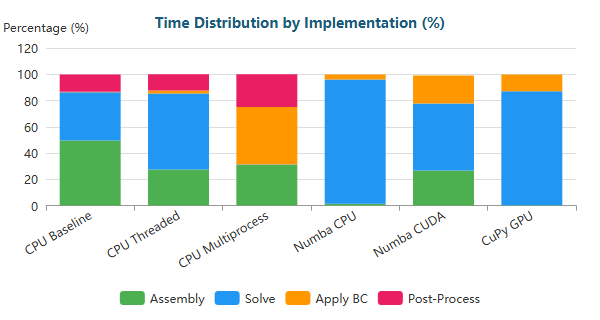
**Time Distribution:**



Pie



Pie

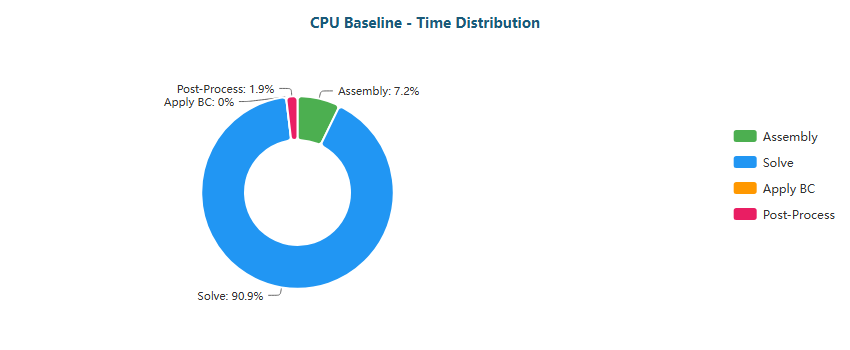


Stacked Bar

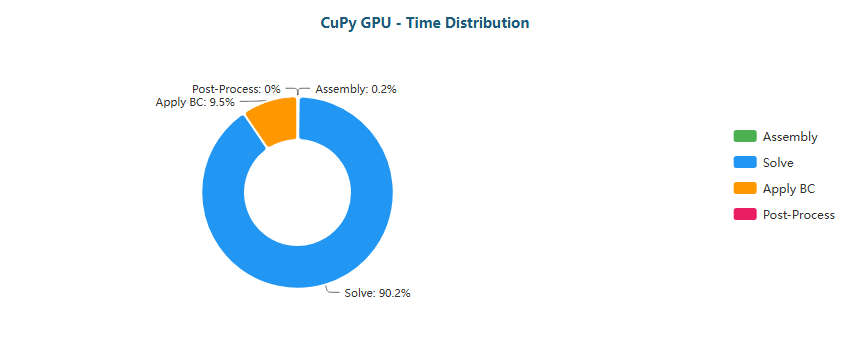
#### T-Junction (L) - 768,898 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (91%) | Assembly (7%) |
| CPU Threaded | Solve (72%) | Assembly (18%) |
| CPU Multiprocess | BC (77%) | Assembly (14%) |
| Numba CPU | Solve (100%) | BC (0%) |
| Numba CUDA | Solve (54%) | Assembly (25%) |
| CuPy GPU | Solve (90%) | BC (10%) |

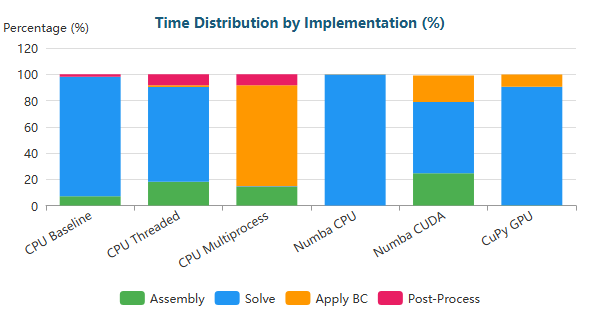
**Time Distribution:**



Pie



Pie

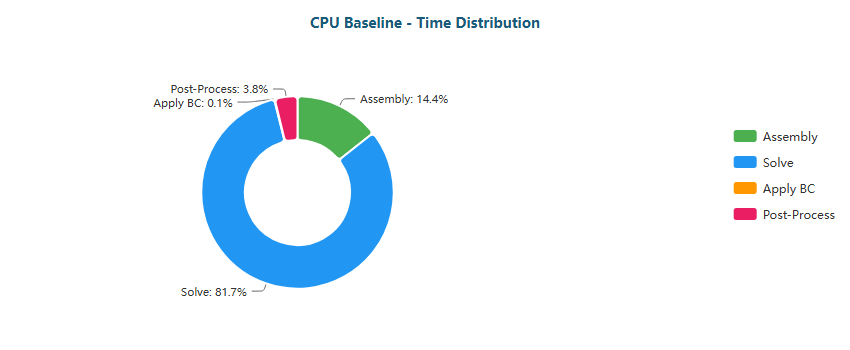


Stacked Bar

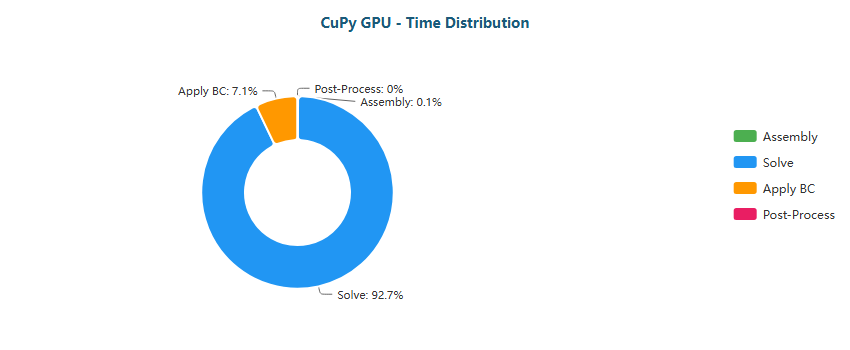
#### T-Junction (XL) - 1,291,289 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (82%) | Assembly (14%) |
| CPU Threaded | Solve (81%) | Assembly (13%) |
| CPU Multiprocess | BC (84%) | Assembly (10%) |
| Numba CPU | Solve (99%) | BC (0%) |
| Numba CUDA | Solve (61%) | Assembly (22%) |
| CuPy GPU | Solve (93%) | BC (7%) |

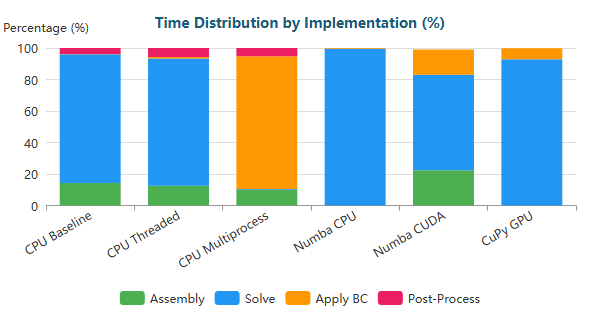
**Time Distribution:**



Pie



Pie

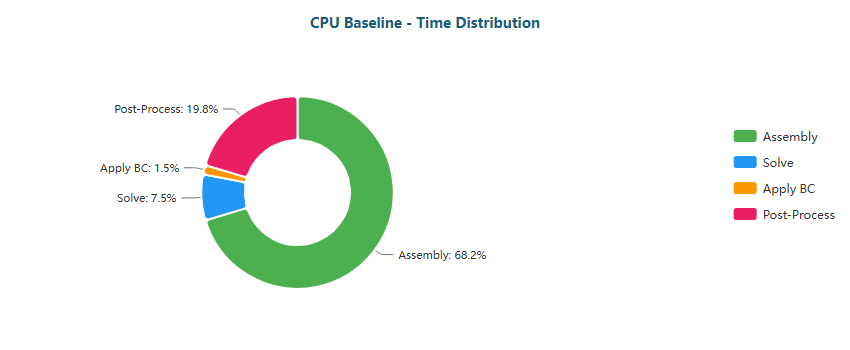


Stacked Bar

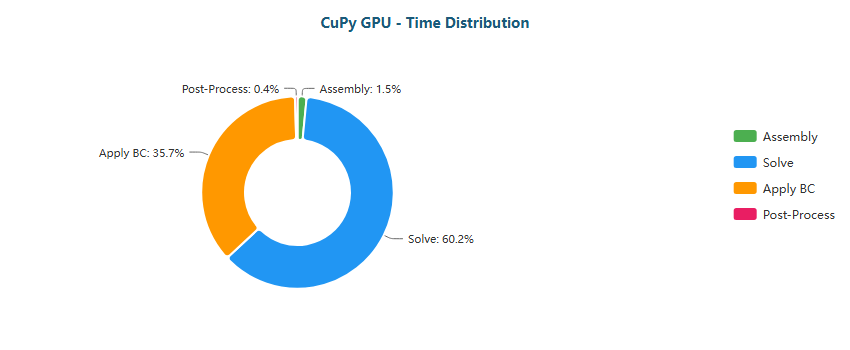
#### Venturi (XS) - 341 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (68%) | Post-Proc (20%) |
| CPU Threaded | Assembly (52%) | Post-Proc (20%) |
| CPU Multiprocess | Post-Proc (60%) | Assembly (40%) |
| Numba CPU | Solve (55%) | BC (19%) |
| Numba CUDA | Solve (88%) | Assembly (6%) |
| CuPy GPU | Solve (60%) | BC (36%) |

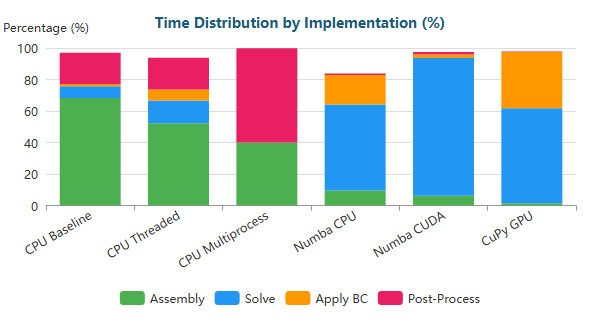
**Time Distribution:**



Pie



Pie

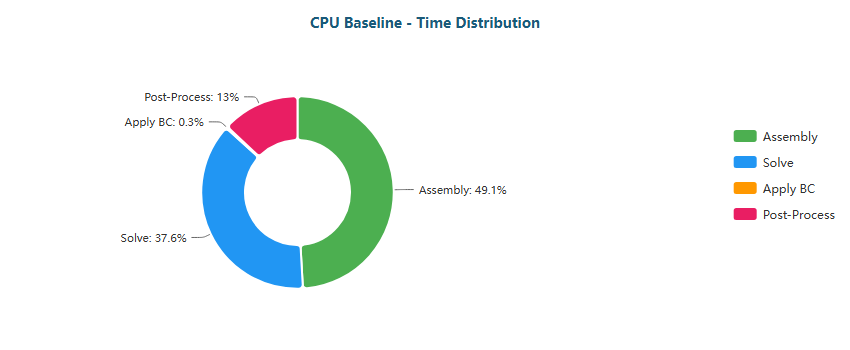


Stacked Bar

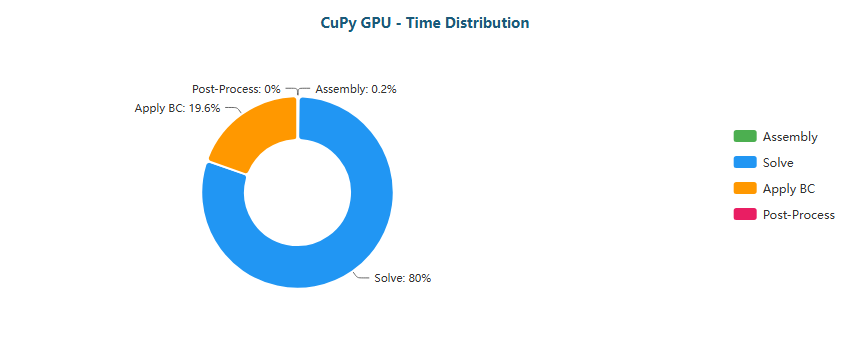
#### Venturi (M) - 194,325 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (49%) | Solve (38%) |
| CPU Threaded | Solve (57%) | Assembly (28%) |
| CPU Multiprocess | BC (58%) | Assembly (23%) |
| Numba CPU | Solve (95%) | BC (4%) |
| Numba CUDA | Solve (50%) | Assembly (28%) |
| CuPy GPU | Solve (80%) | BC (20%) |

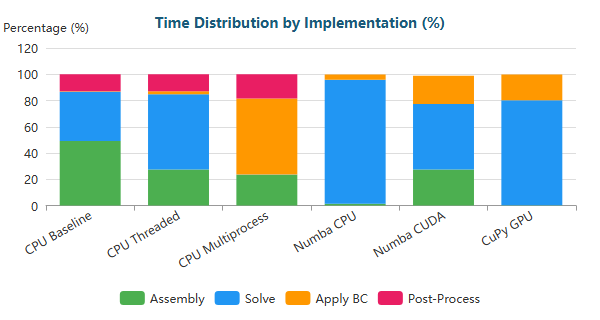
**Time Distribution:**



Pie



Pie

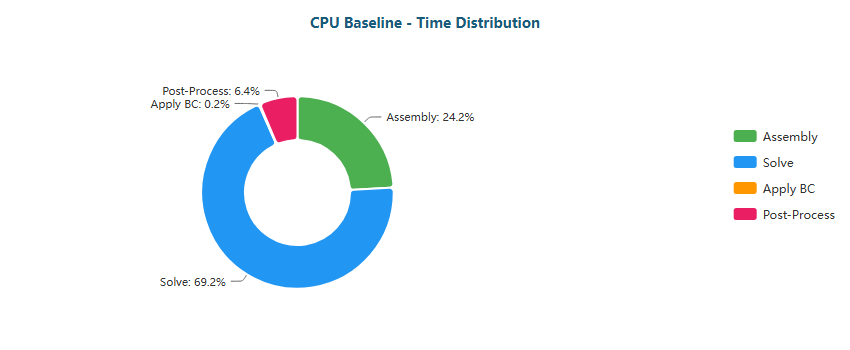


Stacked Bar

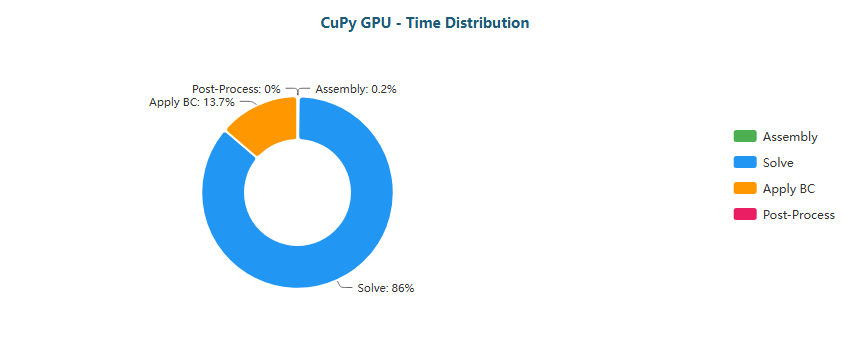
#### Venturi (L) - 763,707 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (69%) | Assembly (24%) |
| CPU Threaded | Solve (71%) | Assembly (19%) |
| CPU Multiprocess | BC (86%) | Assembly (9%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (52%) | Assembly (27%) |
| CuPy GPU | Solve (86%) | BC (14%) |

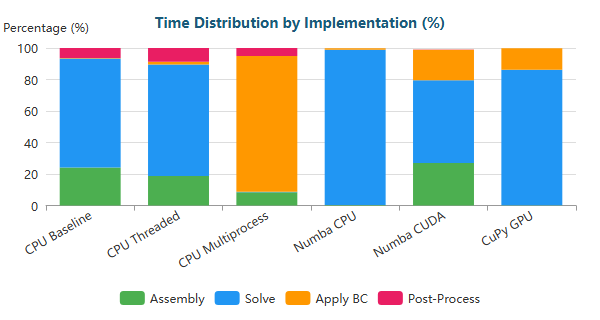
**Time Distribution:**



Pie



Pie

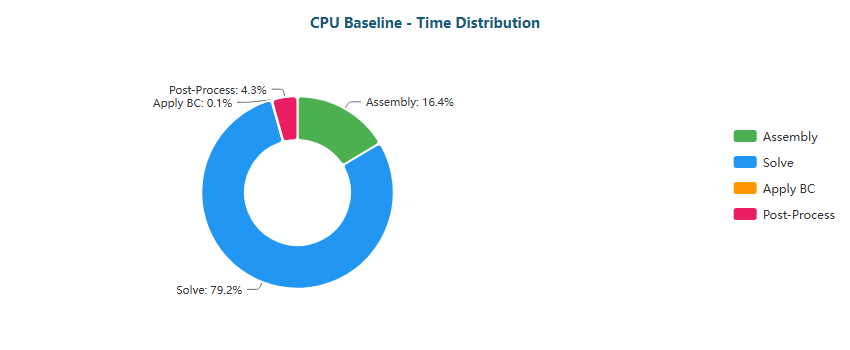


Stacked Bar

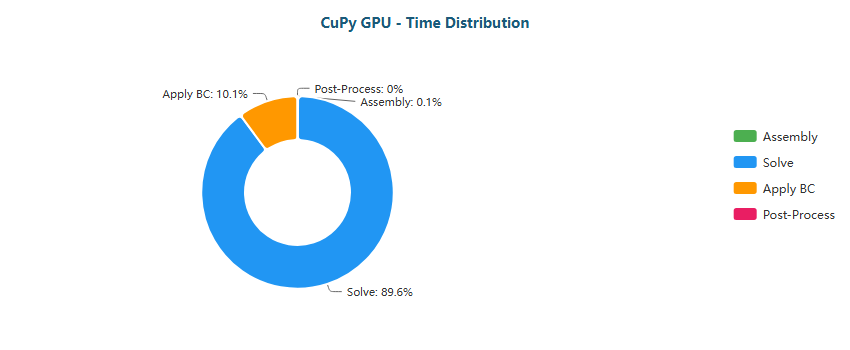
#### Venturi (XL) - 1,284,412 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (79%) | Assembly (16%) |
| CPU Threaded | Solve (79%) | Assembly (13%) |
| CPU Multiprocess | BC (91%) | Assembly (6%) |
| Numba CPU | Solve (99%) | BC (1%) |
| Numba CUDA | Solve (57%) | Assembly (24%) |
| CuPy GPU | Solve (90%) | BC (10%) |

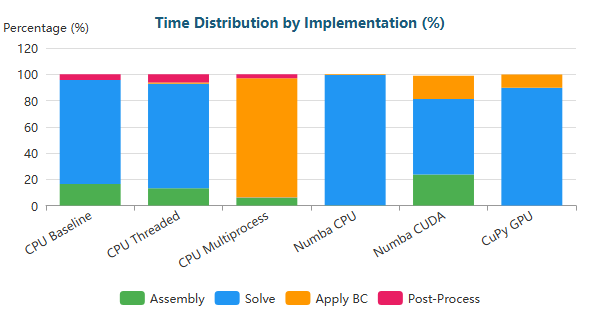
**Time Distribution:**



Pie



Pie

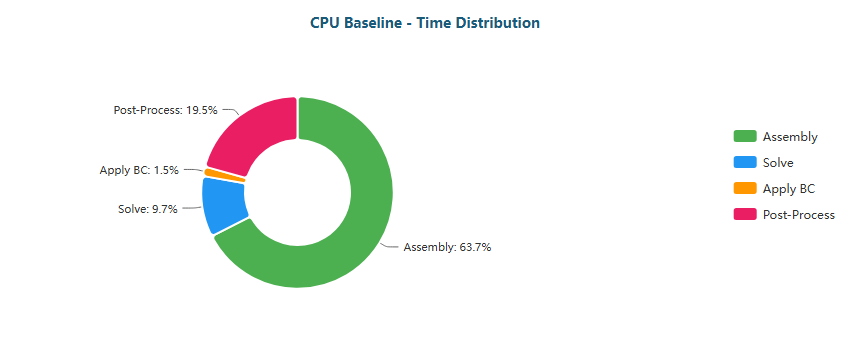


Stacked Bar

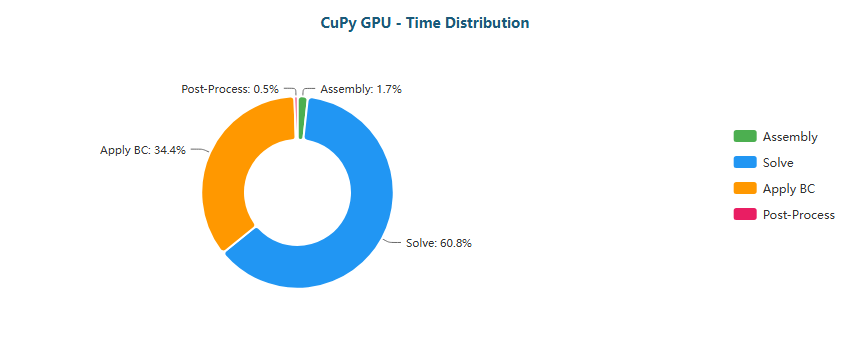
#### Y-Shaped (XS) - 201 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (64%) | Post-Proc (19%) |
| CPU Threaded | Assembly (47%) | Post-Proc (21%) |
| CPU Multiprocess | Assembly (50%) | Post-Proc (50%) |
| Numba CPU | Solve (58%) | BC (13%) |
| Numba CUDA | Solve (87%) | Assembly (6%) |
| CuPy GPU | Solve (61%) | BC (34%) |

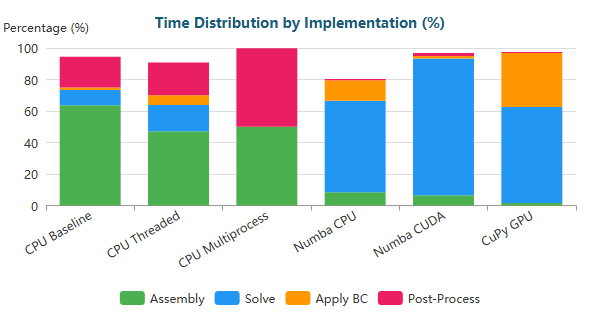
**Time Distribution:**



Pie



Pie



Stacked Bar

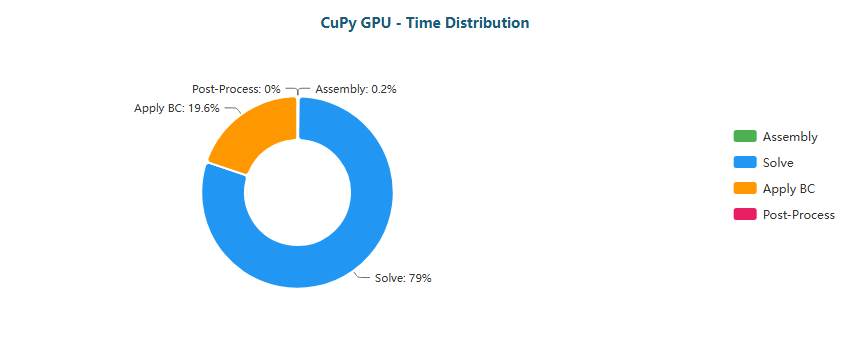
#### Y-Shaped (M) - 195,853 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Assembly (47%) | Solve (40%) |
| CPU Threaded | Solve (61%) | Assembly (25%) |
| CPU Multiprocess | BC (38%) | Assembly (33%) |
| Numba CPU | Solve (93%) | BC (5%) |
| Numba CUDA | Solve (49%) | BC (26%) |
| CuPy GPU | Solve (79%) | BC (20%) |

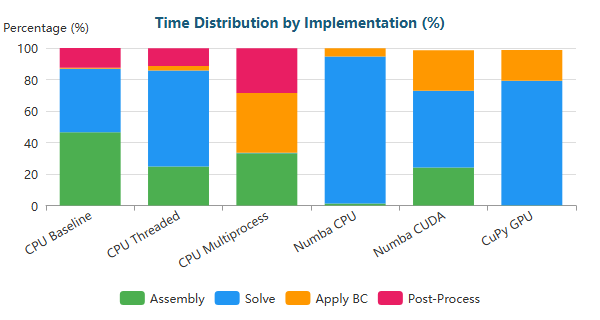
**Time Distribution:**



Pie



Pie

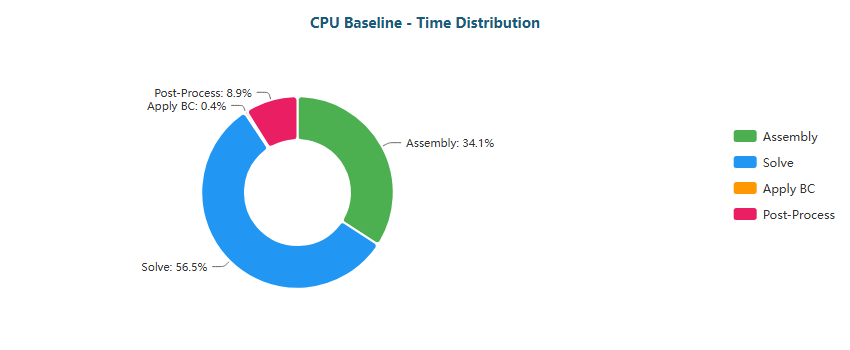


Stacked Bar

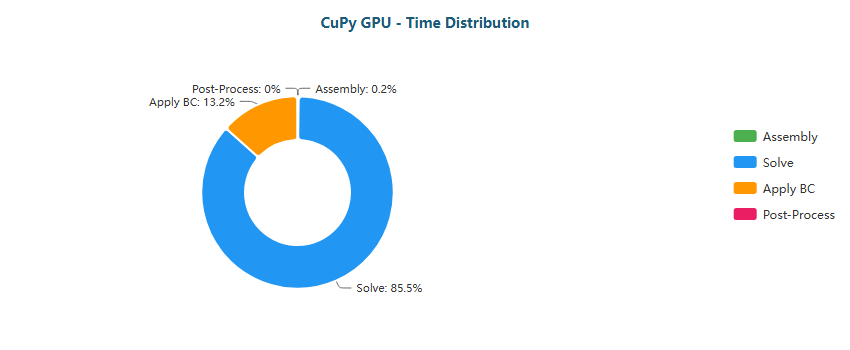
#### Y-Shaped (L) - 772,069 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (57%) | Assembly (34%) |
| CPU Threaded | Solve (75%) | Assembly (16%) |
| CPU Multiprocess | BC (75%) | Assembly (15%) |
| Numba CPU | Solve (97%) | BC (3%) |
| Numba CUDA | Solve (52%) | Assembly (23%) |
| CuPy GPU | Solve (86%) | BC (13%) |

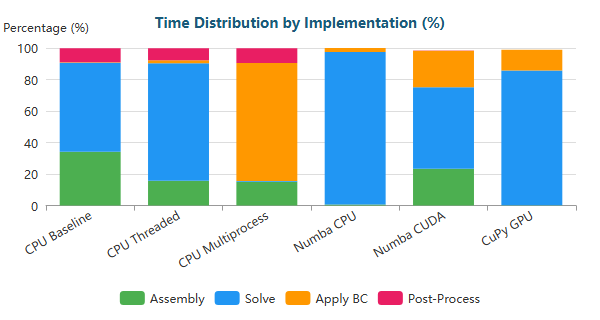
**Time Distribution:**



Pie



Pie

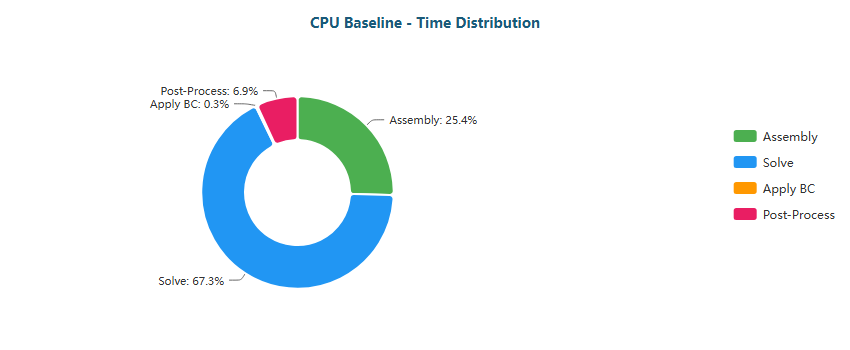


Stacked Bar

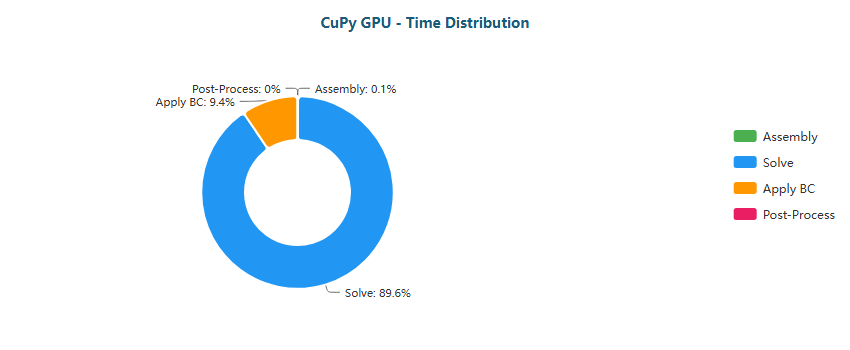
#### Y-Shaped (XL) - 1,357,953 nodes

| Implementation | Primary Bottleneck | Secondary Bottleneck |
| --- | --- | --- |
| CPU Baseline | Solve (67%) | Assembly (25%) |
| CPU Threaded | Solve (82%) | Assembly (12%) |
| CPU Multiprocess | BC (82%) | Assembly (11%) |
| Numba CPU | Solve (98%) | BC (2%) |
| Numba CUDA | Solve (58%) | BC (20%) |
| CuPy GPU | Solve (90%) | BC (9%) |

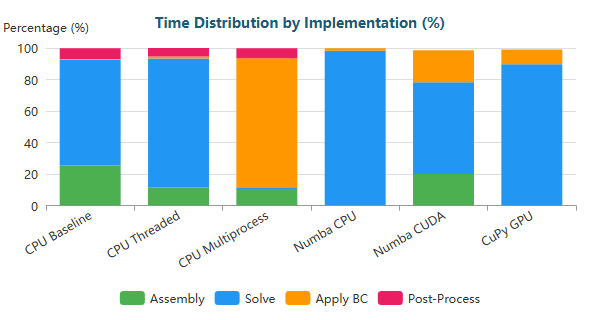
**Time Distribution:**



Pie



Pie



Stacked Bar

### Why Each Optimization Helps

| Transition | Reason |
| --- | --- |
| Baseline → Threaded | Limited by Python GIL; threads only help for I/O |
| Threaded → Multiprocess | Bypasses GIL via separate processes; IPC overhead limits gains |
| Multiprocess → Numba CPU | JIT compilation eliminates interpreter overhead; true parallel loops |
| Numba CPU → Numba CUDA | GPU parallelism: thousands of threads vs dozens of CPU cores |
| Numba CUDA → CuPy GPU | CUDA C kernels more optimized than Numba-generated PTX |

#### 1.4.2.1 Bottleneck Migration Pattern on RTX 5060 Ti

Across all five geometries (Backward-Facing Step, Elbow, S-Bend, T-Junction, Venturi, Y-Shaped), the RTX 5060 Ti shows a stable migration sequence:

* **CPU Baseline → CPU Threaded**
  + **XS meshes:** Assembly remains dominant (≈64–72%), Post-Processing stays high (≈18–26%).
    - BFS-XS: Assembly ≈69%, Post-Proc ≈19%
    - Elbow-XS: Assembly ≈70%, Post-Proc ≈20%
    - T-Junction-XS: Assembly ≈72%, Post-Proc ≈18%
  + Threading reduces wall-time but **does not change the dominant stage**, confirming interpreter-level constraints.
* **CPU Multiprocess**
  + **XS meshes:** overhead-driven behavior with **Post-Proc or 50/50 splits**
    - BFS-XS / Venturi-XS: Post-Proc ≈60%, Assembly ≈40%
    - Elbow-XS / S-Bend-XS / T-Junction-XS / Y-XS: ≈50/50 Assembly vs Post-Proc
  + **M/L/XL meshes:** clear **BC-dominated regime**
    - BFS-M: BC ≈53%
    - Elbow-M: BC ≈69%
    - Venturi-M: BC ≈58%
    - L/XL across geometries: BC ≈75–94%
  + Interpretation: process orchestration and memory duplication cause **Apply BC to become the dominant tax**, enforcing a hard scaling ceiling.
* **Numba CPU (JIT)**
  + Bottleneck transitions to **Solve** almost universally:
    - M/L/XL: Solve ≈93–100%
    - XS: Solve ≈54–61%, with **BC visible** (≈13–19%)
  + Confirms that JIT removes Python overhead, exposing true numerical cost.
* **Numba CUDA**
  + **XS:** Solve dominates (≈87–90%)
  + **M/L/XL:** stable **Solve–Assembly split**
    - Solve ≈47–64%
    - Assembly ≈20–30%
    - BC ≈15–23%
  + Interpretation: GPU execution exposes **assembly as a memory-bound stage** rather than eliminating it.
* **CuPy GPU**
  + **M/L/XL:** solver-dominated regime
    - Solve ≈80–94%
    - BC ≈6–20%
  + **XS:** BC becomes unusually large
    - Solve ≈60–71%
    - BC ≈25–36%
  + Indicates inability to amortize launch/sync + BC enforcement at tiny scales.

**Core takeaway:** on RTX 5060 Ti, effective GPU usage leads to a **solver-bound pipeline**, with **Apply BC as the persistent secondary cost**, while **CPU multiprocess collapses into BC dominance at scale**.

#### 1.4.2.2 Optimization Implications and Practical Limits on RTX 5060 Ti

The RTX 5060 Ti results define clear optimization boundaries:

* **XS meshes**
  + Avoid multiprocess (overhead-dominated).
  + GPU viable but BC-heavy.
  + **Best choice:** Numba JIT CPU.
* **M meshes**
  + CPU multiprocess becomes a **BC trap**.
  + Numba CPU already near-solve-limited.
  + CuPy GPU cleanly solver-dominated.
  + **Best choice:** CuPy GPU (or Numba CPU if GPU unavailable).
* **L/XL meshes**
  + Multiprocess collapses into extreme BC dominance (≈75–94%).
  + CuPy reaches solver ceiling (Solve ≈86–94%).
  + **Best choice:** CuPy GPU.
* **Remaining optimization headroom**
  + **BC fusion/minimization** (critical at XS/M on GPU).
  + **Assembly optimization** for Numba CUDA (reduce scatter, improve coalescing).
  + **Solver-level improvements** dominate ultimate ceiling (preconditioning, iteration reduction).

Overall, the RTX 5060 Ti behaves as a **solver-dominated GPU at scale**, with pronounced **BC sensitivity** at XS and a uniquely severe **multiprocess penalty** at M/L/XL.

## 2. Cross-Platform Comparative Analysis

This section consolidates the benchmark results presented in Sections 4.5-4.7 into a unified comparative analysis.  
Rather than reiterating individual measurements, the focus here is on **interpreting performance trends**, **explaining architectural effects**, and **extracting general conclusions** regarding execution models and GPU classes.

### 2.1 CPU vs GPU: Where the Paradigm Shifts

Across all geometries and medium-to-large meshes, a clear and consistent transition point emerges:

* **Small meshes (XS)**  
  GPU execution is systematically slower than optimized CPU variants due to:
  + kernel launch overhead,
  + PCIe latency,
  + underutilization of GPU parallelism.
* **Medium meshes (M)**  
  GPU acceleration becomes dominant, with speedups ranging from:
  + **~11× (RTX 5060 Ti)**
  + **~20-40× (RTX 4090)**
  + **~30-60× (RTX 5090)**

This confirms that GPU acceleration is not universally beneficial, but **highly problem-size dependent**.

### 2.2 CPU Scaling Limits

The benchmark reveals well-defined limits for CPU-based optimization strategies.

| CPU Strategy | Observed Benefit | Limiting Factor |
| --- | --- | --- |
| Threading | 1.2× - 2.1× | Python GIL |
| Multiprocessing | 1.5× - 2.7× | IPC overhead |
| Numba JIT | 2× - 6× | Memory bandwidth |

Even with aggressive JIT compilation, **CPU performance saturates early**.  
For medium meshes, the solver becomes:

* **memory-bound**, and
* dominated by **sparse matrix-vector products**.

This explains why Numba CPU converges to similar performance as multiprocessing for large problems.

### 2.3 GPU Acceleration: Numba CUDA vs CuPy RawKernel

A consistent hierarchy is observed across all GPUs:

| GPU Execution Model | Characteristics | Performance |
| --- | --- | --- |
| Numba CUDA | Python-defined kernels, easier development | High |
| CuPy RawKernel | Native CUDA C, full control | Highest |

Key observations:

* **CuPy GPU consistently outperforms Numba CUDA** for medium meshes.
* Gains range from **1.3× to 1.8×** over Numba CUDA.
* The advantage increases with:
  + mesh size,
  + solver dominance,
  + memory bandwidth pressure.

This confirms that **kernel maturity and low-level control matter** once GPU execution becomes solver-bound.

### 2.4 Cross-GPU Performance Scaling

A core objective of this benchmark was to separate **software scaling** from **hardware scaling**.

#### Aggregate Speedup (Medium Meshes)

| GPU | Typical Speedup vs CPU Baseline |
| --- | --- |
| RTX 5060 Ti | 11× - 18× |
| RTX 4090 | 20× - 42× |
| RTX 5090 | 25× - 60× |

However, performance does **not** scale linearly with theoretical FLOPs.

#### Interpretation

* The FEM solver is **memory-bandwidth dominated**, not compute-bound.
* Higher-end GPUs benefit from:
  + larger L2 cache,
  + higher memory throughput,
  + better latency hiding.
* The RTX 5090 advantage is strongest for:
  + CG-heavy cases,
  + large sparse matrices,
  + solver-dominated workloads.

This confirms that **architectural balance**, not raw FLOPs, drives FEM performance.

### 2.5 Bottleneck Evolution Across Platforms

A central insight from the benchmark is the **systematic migration of bottlenecks**:

| Execution Stage | CPU Baseline | Numba CPU | GPU (CuPy) |
| --- | --- | --- | --- |
| Assembly | Dominant | Minor | Negligible |
| Solve | Secondary | Dominant | Overwhelming |
| Apply BC | Minor | Minor | Non-negligible |
| Post-processing | Visible | Minimal | Negligible |

Key implications:

* GPU acceleration **eliminates assembly as a bottleneck**.
* The **linear solver dominates runtime** in all optimized variants.
* On GPU, boundary condition application becomes visible due to:
  + atomic operations,
  + irregular memory access,
  + limited arithmetic intensity.

This validates the design decision to prioritize GPU-resident solvers.

### 2.6 Efficiency vs Absolute Performance

While the RTX 5090 delivers the highest absolute performance, efficiency considerations are relevant:

| GPU | Relative Performance | Cost / Power Consideration |
| --- | --- | --- |
| RTX 5060 Ti | Moderate | High efficiency per cost |
| RTX 4090 | Very high | Balanced performance |
| RTX 5090 | Extreme | Diminishing returns |

For production environments, this suggests:

* **Mid-range GPUs** are sufficient for moderate FEM workloads.
* **High-end GPUs** are justified for:
  + very large meshes,
  + repeated simulations,
  + solver-dominated pipelines.

### 2.7 Robustness and Numerical Consistency

Crucially, acceleration does **not** alter numerical behavior:

* Identical CG iteration counts across platforms.
* Consistent residual norms at convergence.
* No divergence or fallback behavior observed.

This confirms that performance gains are achieved **without sacrificing numerical correctness**.

### 2.8 Consolidated Summary

| Aspect | Key Conclusion |
| --- | --- |
| CPU optimization | Quickly saturates |
| GPU benefit | Strongly size-dependent |
| Best execution model | CuPy RawKernel |
| Dominant bottleneck | Sparse solver |
| Best scaling factor | Memory bandwidth |
| Best overall GPU | RTX 5090 |
| Best cost-efficiency | RTX 5060 Ti |

### 2.9 Final Insights

The benchmark demonstrates that **GPU acceleration fundamentally changes the performance landscape of FEM solvers**, but only when:

* The problem size is sufficiently large,
* Data remains resident on the GPU,
* Solver execution dominates the pipeline.

Beyond this point, performance becomes a function of **memory architecture rather than algorithmic complexity**, placing modern GPUs at a decisive advantage over CPUs for large-scale finite element simulations.