

SVG Images Collection

Generated Script

August 14, 2024

Contents

1	Discussion	1
1.1	Model Characteristics and Performance	1
1.2	Configuration and Platform Considerations	1
1.3	Performance Factors and Optimization	1
1.3.1	Event Processing Time	1
1.4	Performance Optimization: Rollbacks and State Saving	2
1.4.1	Introduction to Rollbacks and State Saving	2
1.4.2	State Saving and Rollback Relationship	2
1.5	Branch Implementations	2
1.5.1	Master Branch	2
1.5.2	Fossil Branch	2
1.5.3	SIMD Branch	3
1.5.4	Hashing Branch	3
1.5.5	Relaxed Compare Branch	3
1.5.6	Unified Queue Branch	3
1.6	GVT Implementation Comparison	4
1.7	Future Work	4
2	Label Definitions	5

3 fossil	6
4 fossillocal	18
5 gaurav_unified	30
6 gaurav_unified_local	42
7 gvt_comparison	51
8 gvt_comparison_local	60
9 hashing	69
10 hashing_local	72
11 SIMD	84
12 SIMD_local	94
13 Top Performers	104

1 Discussion

In this section, we discuss the various aspects of our research on the WARPED simulation kernel, including model characteristics, configuration considerations, and the performance implications of different branch implementations.

1.1 Model Characteristics and Performance

The choice of simulation model significantly influences the performance of the WARPED kernel. We examined several models, including synthetic models like Traffic and PCS, as well as real-world models such as Epidemic-10k-ba and Epidemic-100k-ba [3]. These models vary in their computational requirements and event distribution patterns, affecting the overall simulation performance.

The Epidemic model, for instance, simulates disease outbreak phenomena using a combination of reaction and diffusion processes [2]. It incorporates complex interactions between entities, network structures (e.g., Watts-Strogatz [4] and Barabasi-Albert [1] graphs), and geographical diffusion, making it particularly challenging for parallel discrete event simulation (PDES) systems.

1.2 Configuration and Platform Considerations

While testing proposed changes to the WARPED kernel, it is crucial to determine the scenarios and platforms where they perform optimally. Factors such as instruction sets and specialized hardware can significantly impact performance. As noted earlier, the model itself plays a critical role in determining whether a particular configuration of WARPED works efficiently and quickly.

The debate between GPU and CPU computing for PDES algorithms is ongoing. While GPUs excel at performing simple tasks quickly, CPUs are better suited for handling complex, lengthy computations often required in sophisticated simulations at the Logical Process (LP) level. The choice between platforms depends on the specific requirements of the model and the nature of the computations involved.

1.3 Performance Factors and Optimization

Several factors influence the performance of WARPED simulations:

1.3.1 Event Processing Time

Models like PCS have longer processing times for events at each LP. The grid size, representing the number of LPs, determines the scope of the simulation. It's important to note that the maximum simulation time (max-sim-time) represents the timestamp of the last generated event, not the actual runtime of the simulation.

1.4 Performance Optimization: Rollbacks and State Saving

1.4.1 Introduction to Rollbacks and State Saving

In Parallel Discrete Event Simulation (PDES), the efficiency of rollbacks and state saving mechanisms plays a crucial role in overall performance. This section explores the intricate relationship between state-saving periods and rollback lengths, and presents a mathematical model to optimize these parameters.

1.4.2 State Saving and Rollback Relationship

State saving and rollbacks are interdependent processes in PDES. The frequency of state saving directly impacts the length of potential rollbacks. More frequent state saves reduce the average rollback length but increase overhead, while less frequent saves lead to potentially longer rollbacks but reduce save overhead.

1.5 Branch Implementations

We explored several branch implementations for the WARPED kernel, each focusing on different optimization strategies:

1.5.1 Master Branch

The master branch represents the original implementation of the WARPED 2 kernel. It supports both asynchronous and synchronous Global Virtual Time (GVT) implementations, uses a multiset scheduled queue, and maintains separate processed and unprocessed input queues. This serves as our baseline for comparison with other optimizations.

1.5.2 Fossil Branch

The fossil branch introduces a separate fossil collection thread, eliminating the need for locks during fossil collection. Key features include:

- Tweaks to algorithms to avoid locks during fossil collection
- Potential performance gains on systems without hyperthreading
- Similar runtime to master branch on systems with hyperthreading

Our experiments revealed that simulations with 3+1 threads (3 worker threads and 1 fossil collection thread) resulted in fewer rollbacks. Increasing to 6 threads provided some performance gain, but the improvement was limited due to the increased number of rollbacks. This behavior is likely related to cache swapping delays on our test system with 4 main cores.

1.5.3 SIMD Branch

The SIMD branch aimed to leverage SIMD instructions provided by the AVX platform. However, significant challenges were encountered:

- Requires contiguous memory allocation for members of the Event class
- Necessitates overhauling MPI serialization for multi-node simulations
- Implementation incomplete due to extensive changes required in the WARPED 2 algorithm

While not fully implemented, this branch highlights the potential for SIMD optimizations in future work.

1.5.4 Hashing Branch

The hashing branch implements a technique to reduce the cost of event comparisons:

- Uses hashing for LP names
- One-time hashing cost followed by quick comparisons
- Showed performance gains in experiments
- Potential for further improvement using advanced techniques like Google's MurmurHash

1.5.5 Relaxed Compare Branch

This branch explores a relaxed compare function based on the hypothesis that strict ordering in the scheduled queue is not always necessary:

- Relies on strict ordering at the LP level instead of the global level
- Can fail in certain scenarios
- Showed significant performance gains for some models, particularly those with high LP event processing times

1.5.6 Unified Queue Branch

The unified queue branch implements a combined approach:

- Joins processed and unprocessed events into a single ordered circular queue

- Uses synchronous GVT
- Implements a multiset schedule queue with relaxed ordering
- Utilizes a circular random access iterator for efficient sorting
- Eliminates unnecessary barriers in the simulation

This branch showed promising results by streamlining queue management and reducing synchronization overhead.

1.6 GVT Implementation Comparison

Across our branch implementations, we observed that:

- Synchronous GVT implementation showed better performance on single-node simulations
- Asynchronous GVT performed better on multi-node systems

These findings align with the general industry understanding of GVT types and their optimal use cases.

1.7 Future Work

Future research could explore several promising directions:

- Leveraging new hardware features, such as eco-cores in modern CPUs, for tasks like fossil collection
- Further optimization of the SIMD implementation to overcome current challenges
- Exploration of advanced hashing techniques to improve event comparison efficiency
- Investigating hybrid approaches that combine the strengths of different branch implementations

In conclusion, our research has highlighted the complex interplay between model characteristics, configuration choices, and branch implementations in optimizing the performance of the WARPED simulation kernel. Each branch implementation offers unique advantages and trade-offs, demonstrating the potential for significant improvements in simulation efficiency through targeted optimizations. By carefully considering these factors and applying the most suitable optimizations for specific simulation scenarios, we can continue to enhance the capabilities and performance of the WARPED simulation kernel.

2 Label Definitions

Model	Grid Size	Max Sim Time	Worker Threads	State Save Period	Iterations	GVT Period
Epidemic 10k-ba	10k	20000	3	5, 8, 10	5	1000
Epidemic 100k-ba	100k	6000	3	5, 8, 10	5	1000
PCS 10k	10k	500	3	5, 8, 10	5	1000
Traffic 10k	10k	10000	3	5, 8, 10	5	1000

Table 1: Modified configuration summary for different models

3 fossil

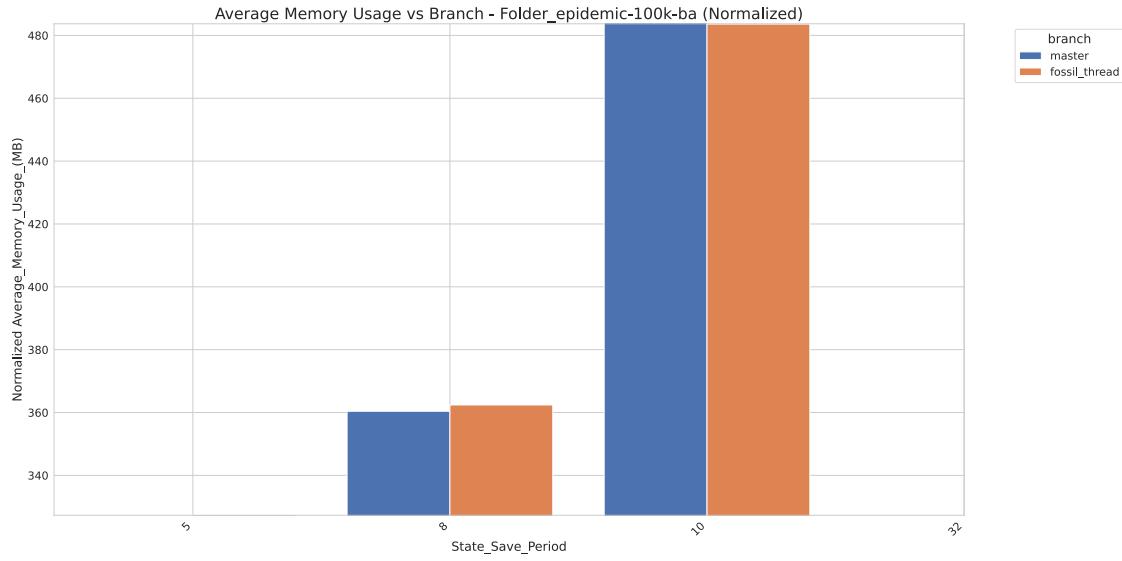


Figure 1: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

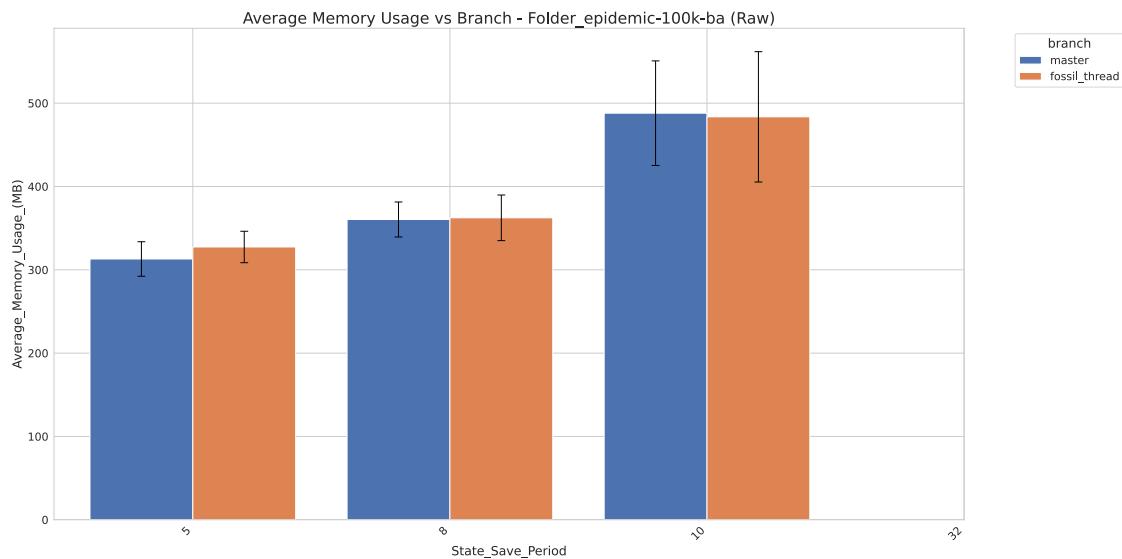


Figure 2: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

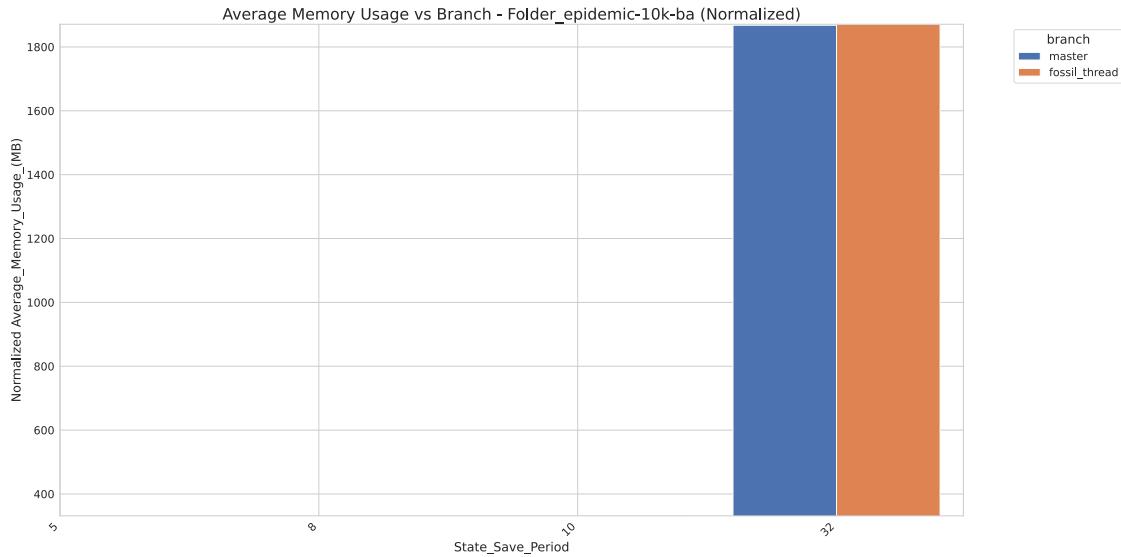


Figure 3: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

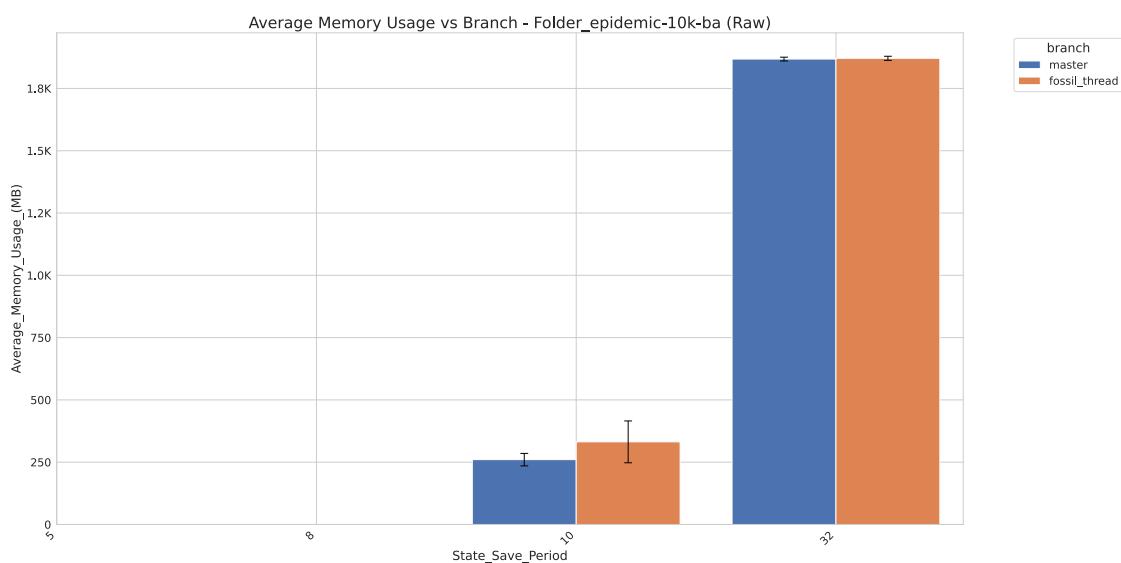


Figure 4: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

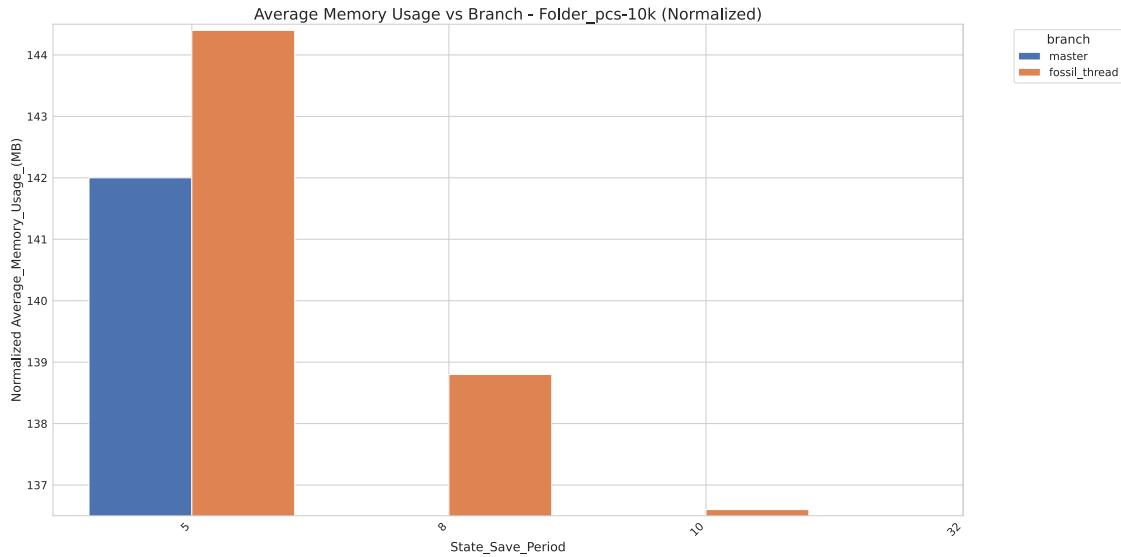


Figure 5: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

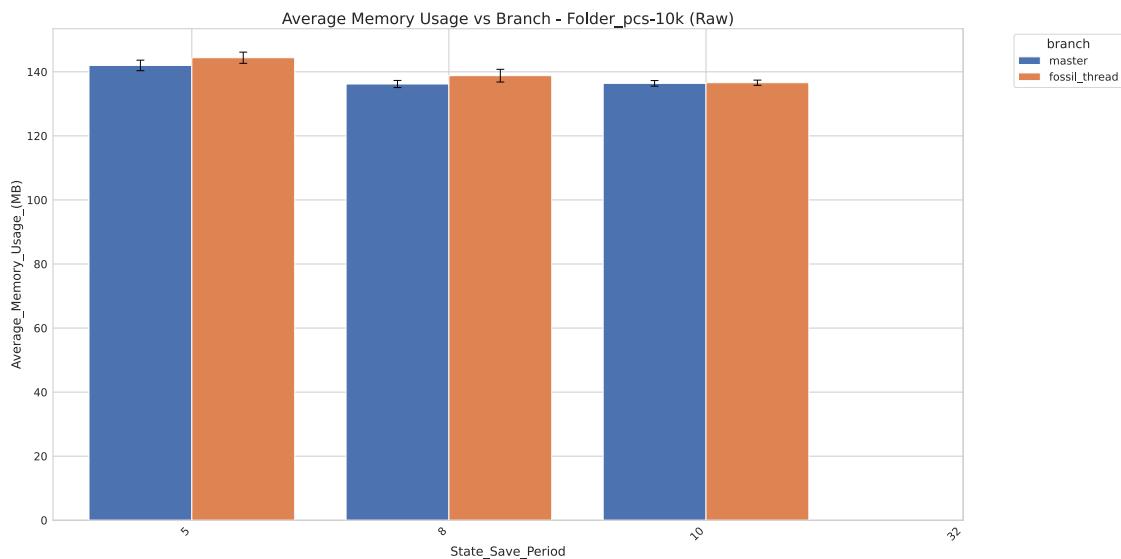


Figure 6: Average Memory Usage vs Branch_Folder_pcs-10k_raw



Figure 7: Average Memory Usage vs Branch_Folder_traffic-10k_normalized

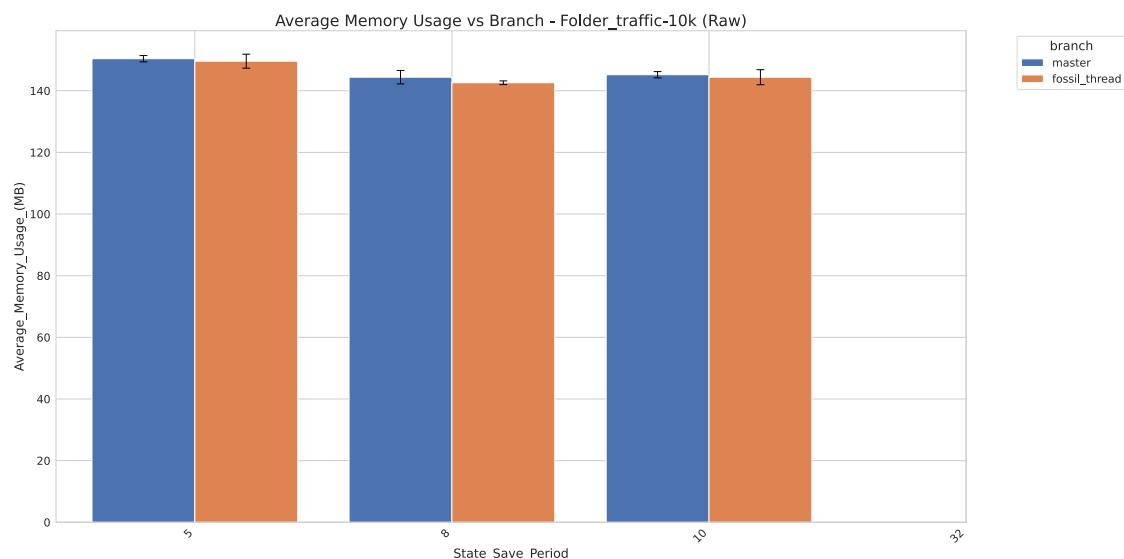


Figure 8: Average Memory Usage vs Branch_Folder_traffic-10k_raw

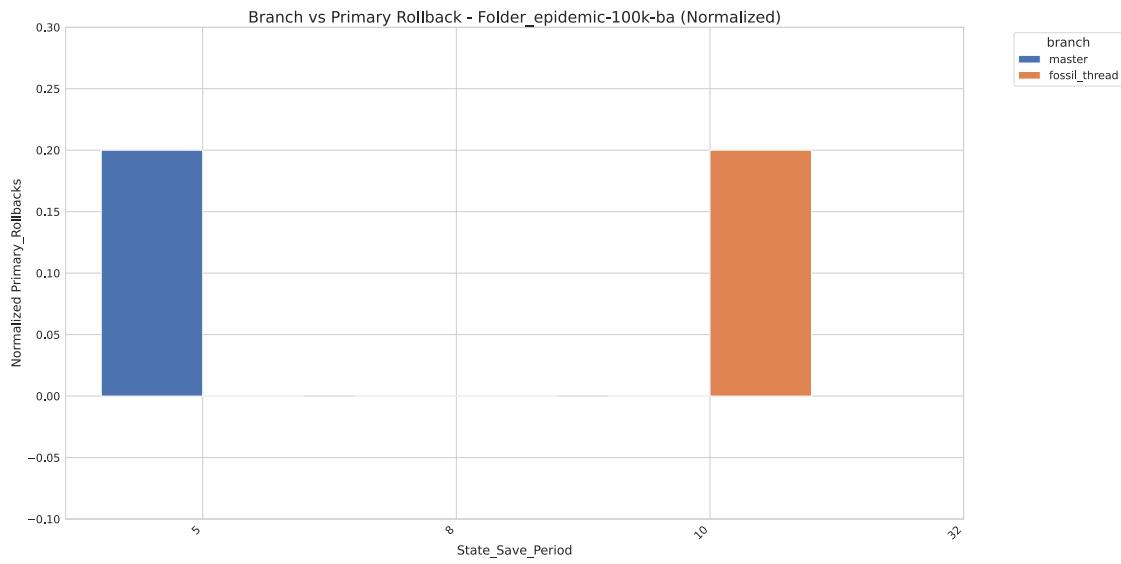


Figure 9: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

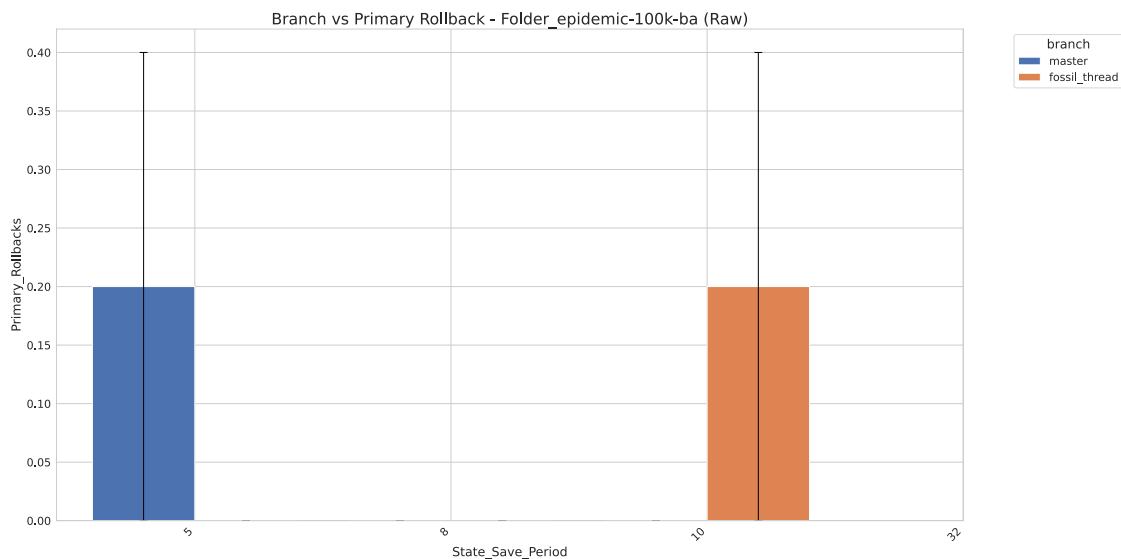


Figure 10: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

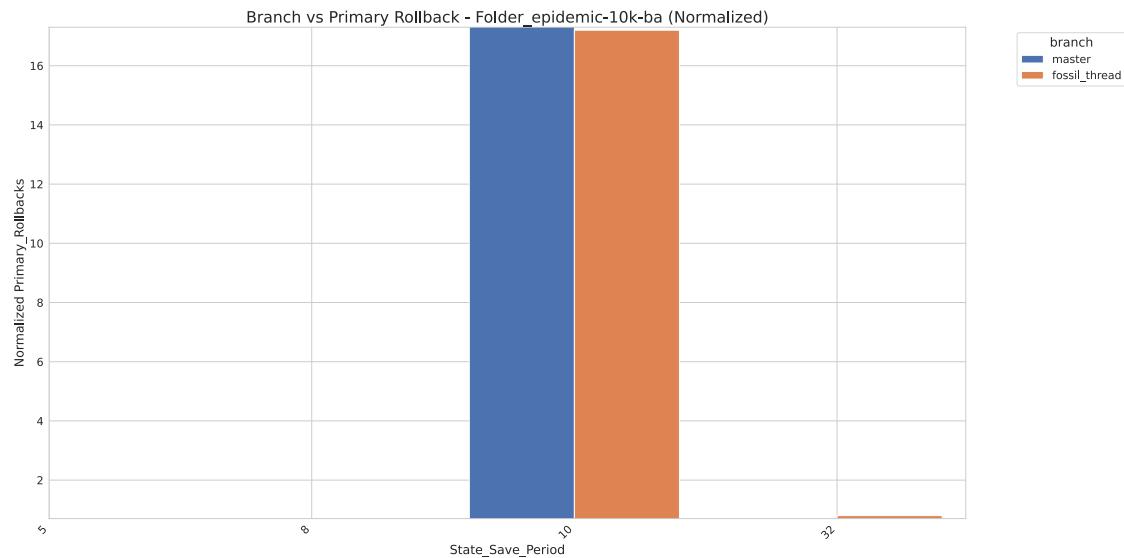


Figure 11: Branch vs Primary Rollback_Folder_epidemic-10k-ba_normalized

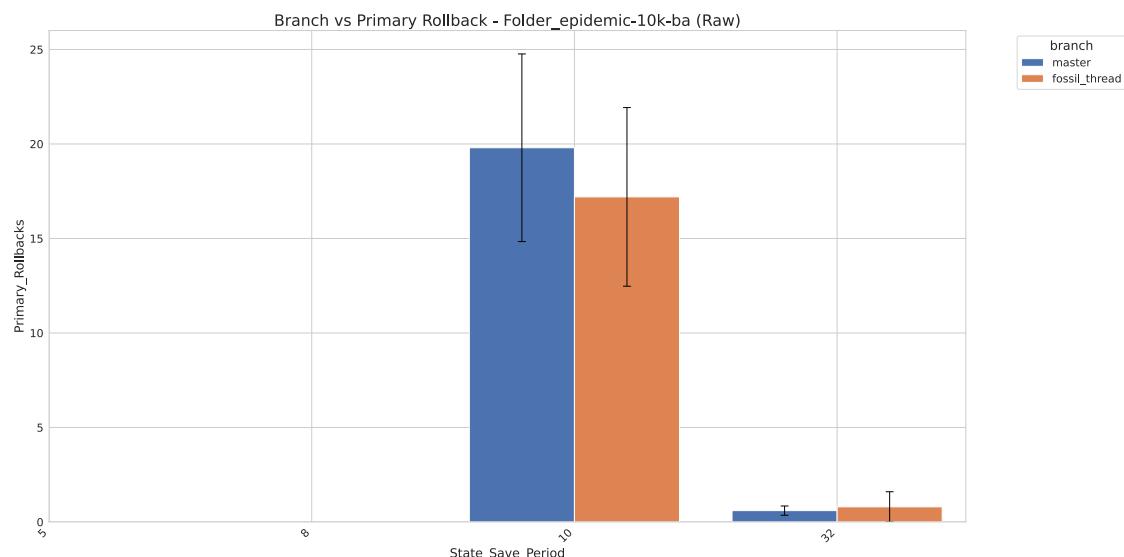


Figure 12: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

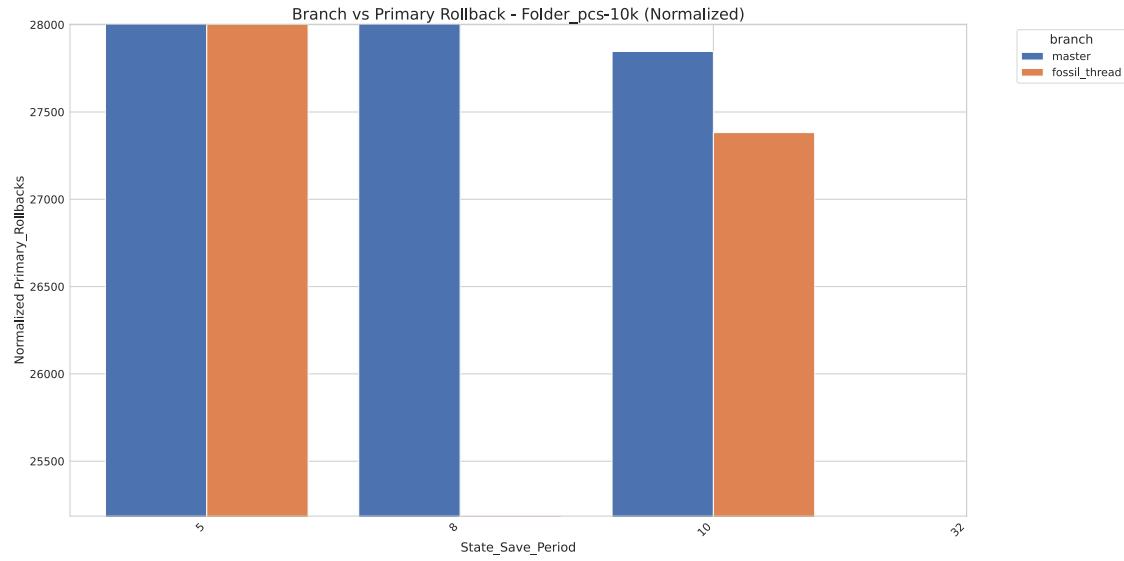


Figure 13: Branch vs Primary Rollback_Folder_pcs-10k_normalized

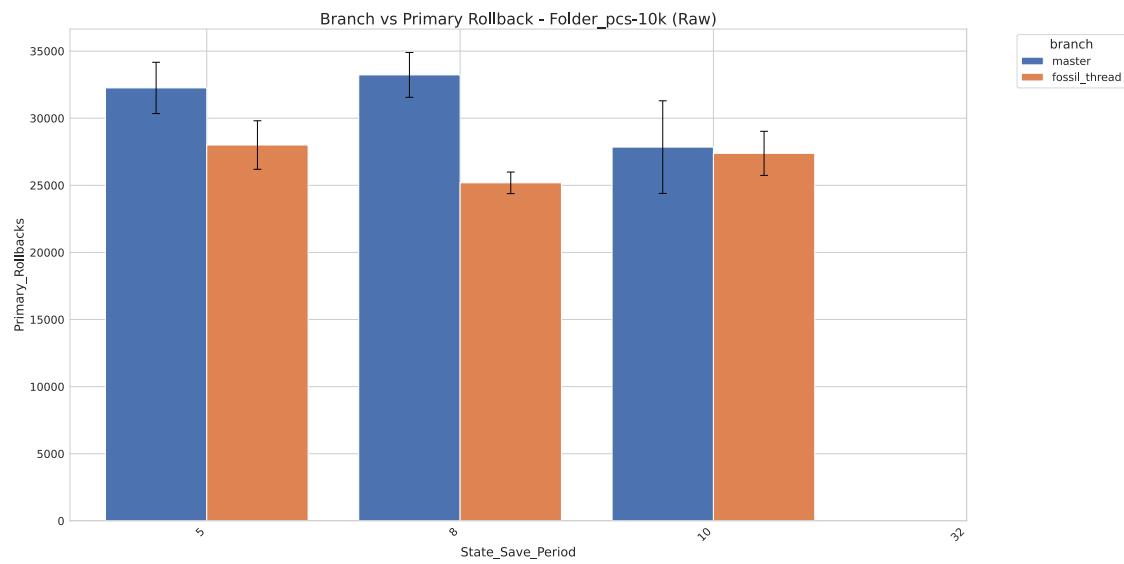


Figure 14: Branch vs Primary Rollback_Folder_pcs-10k_raw

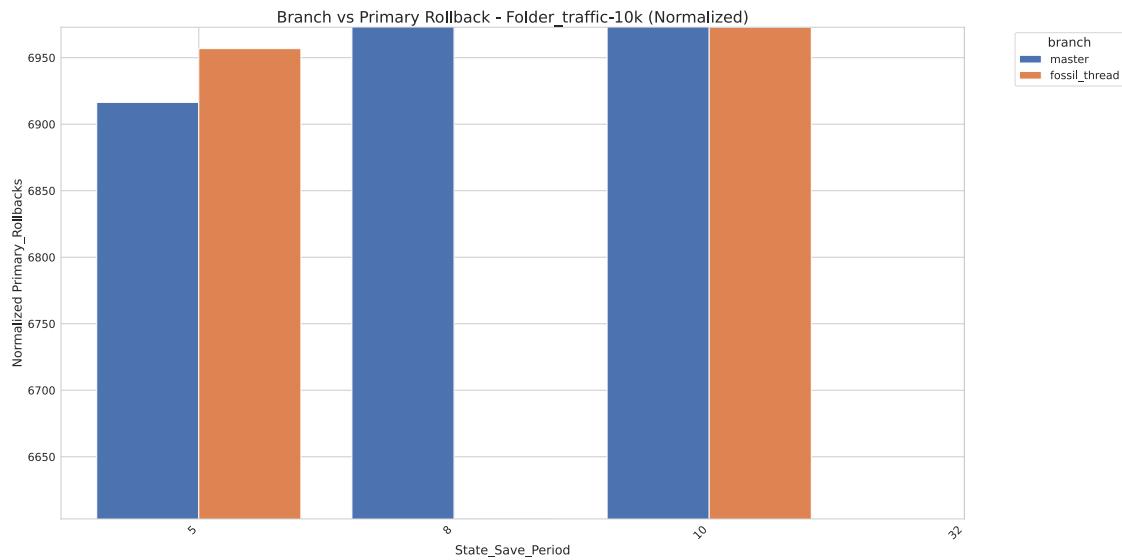


Figure 15: Branch vs Primary Rollback_Folder_traffic-10k_normalized

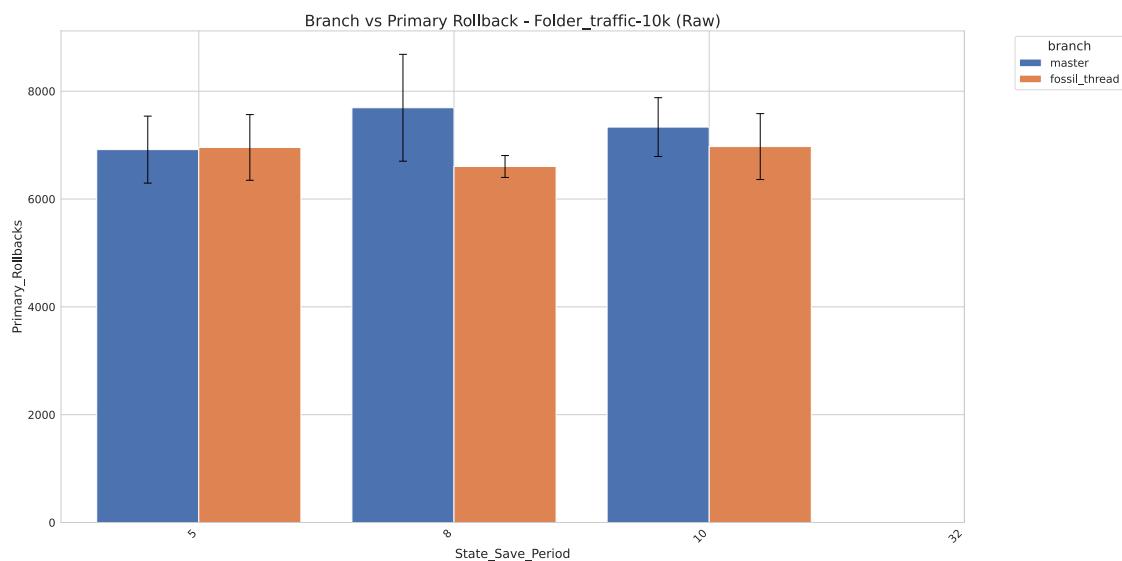


Figure 16: Branch vs Primary Rollback_Folder_traffic-10k_raw

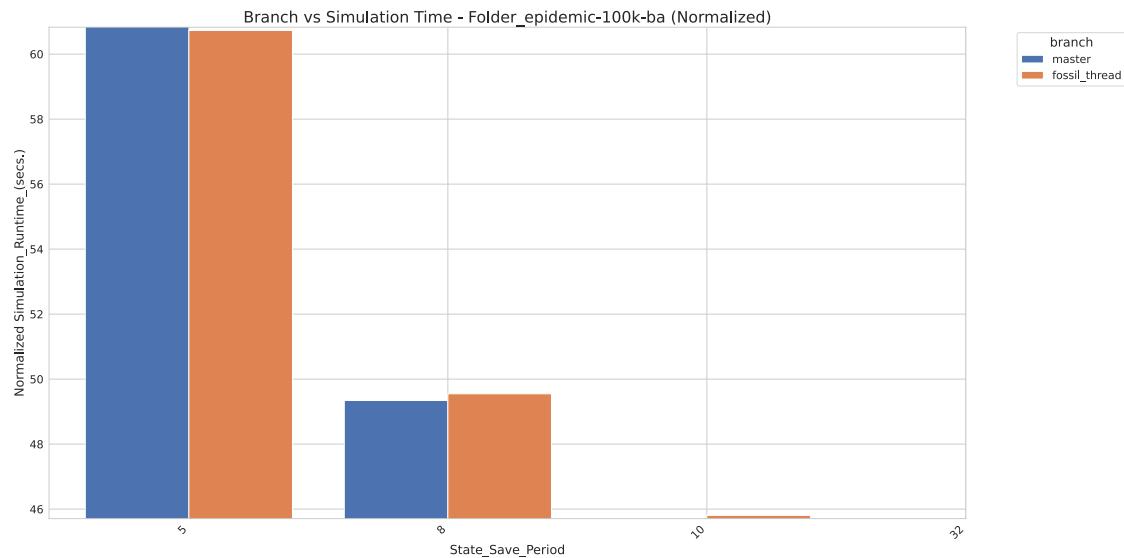


Figure 17: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

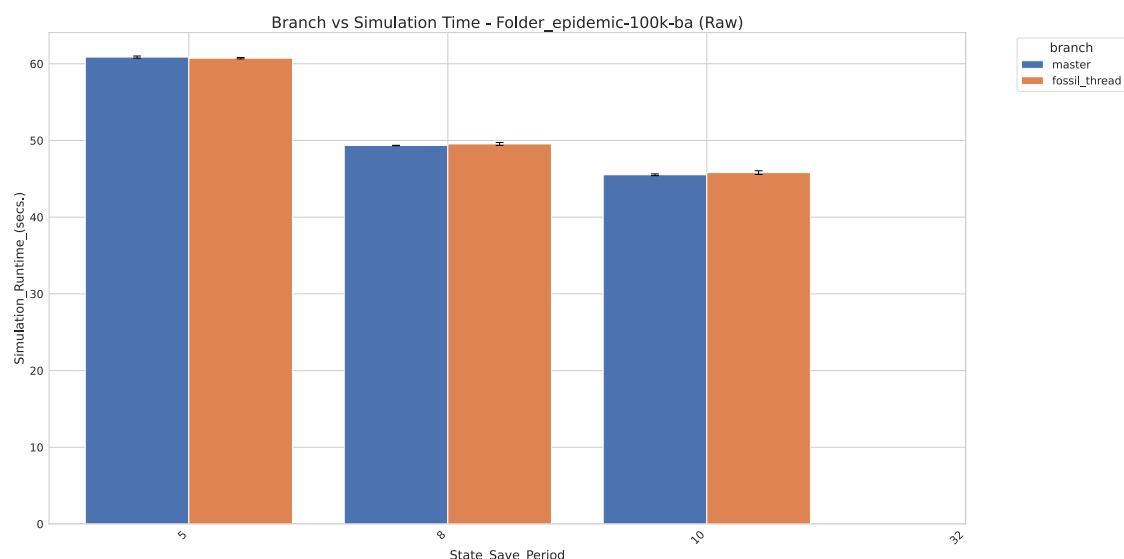


Figure 18: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

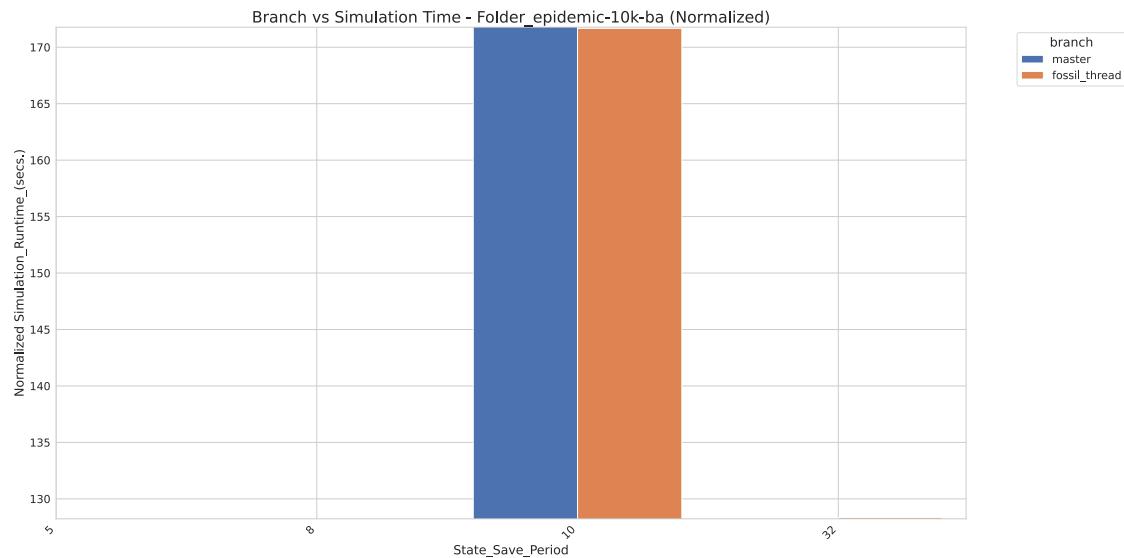


Figure 19: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

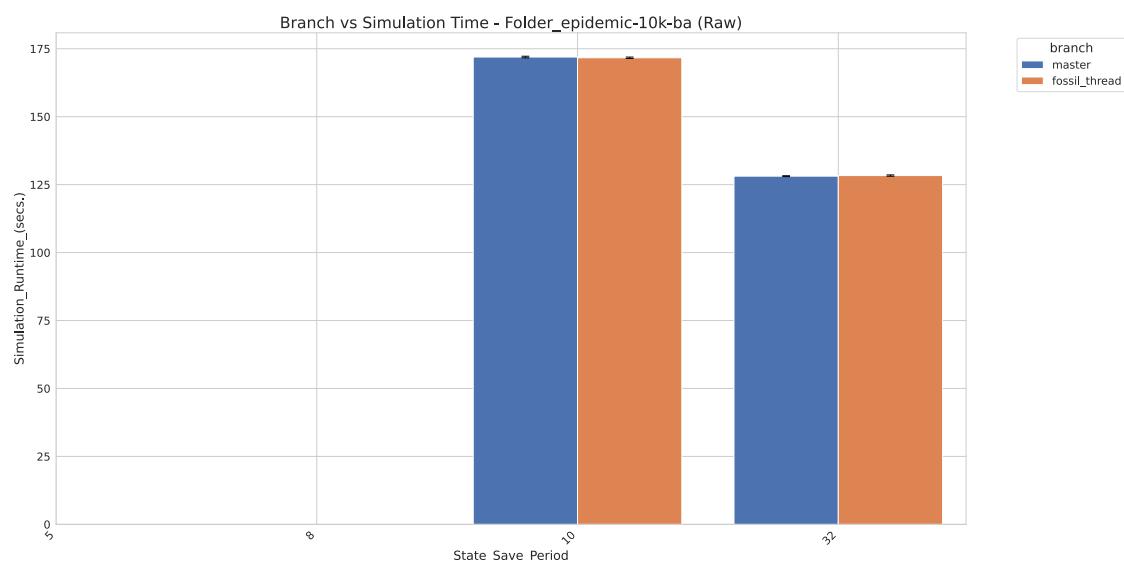


Figure 20: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

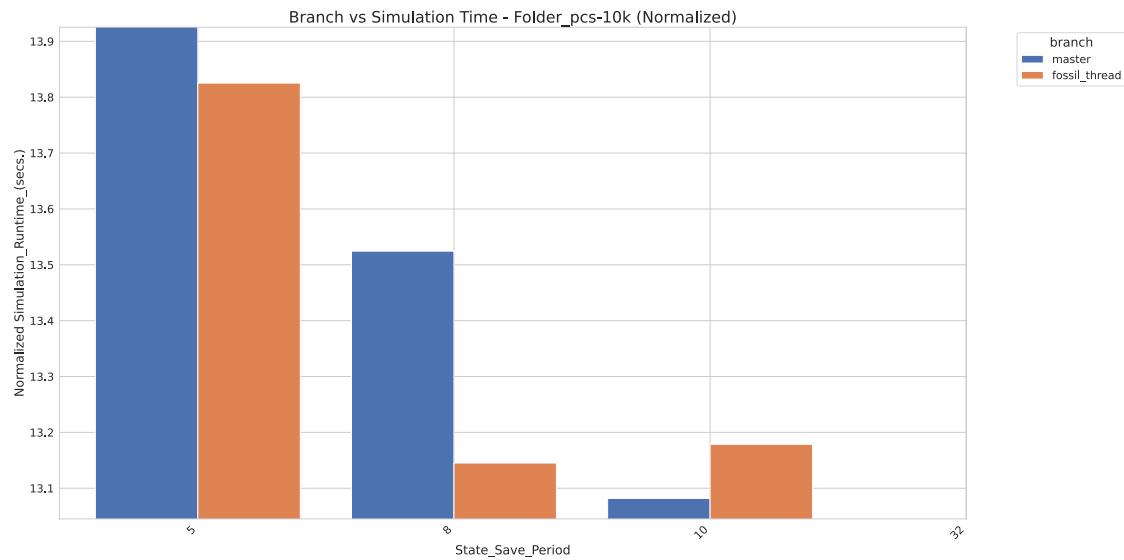


Figure 21: Branch vs Simulation Time_Folder_pcs-10k_normalized

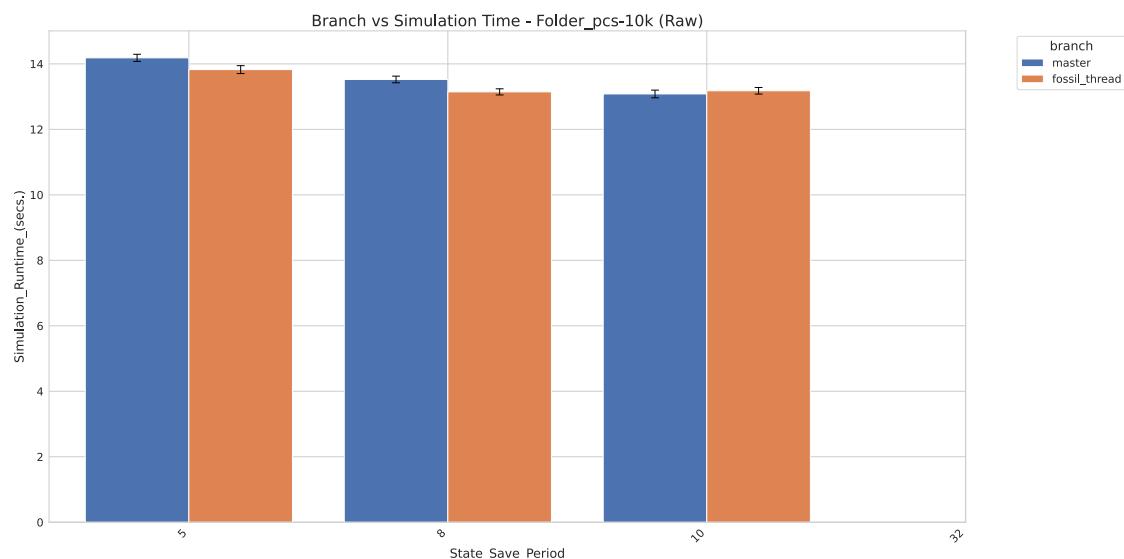


Figure 22: Branch vs Simulation Time_Folder_pcs-10k_raw

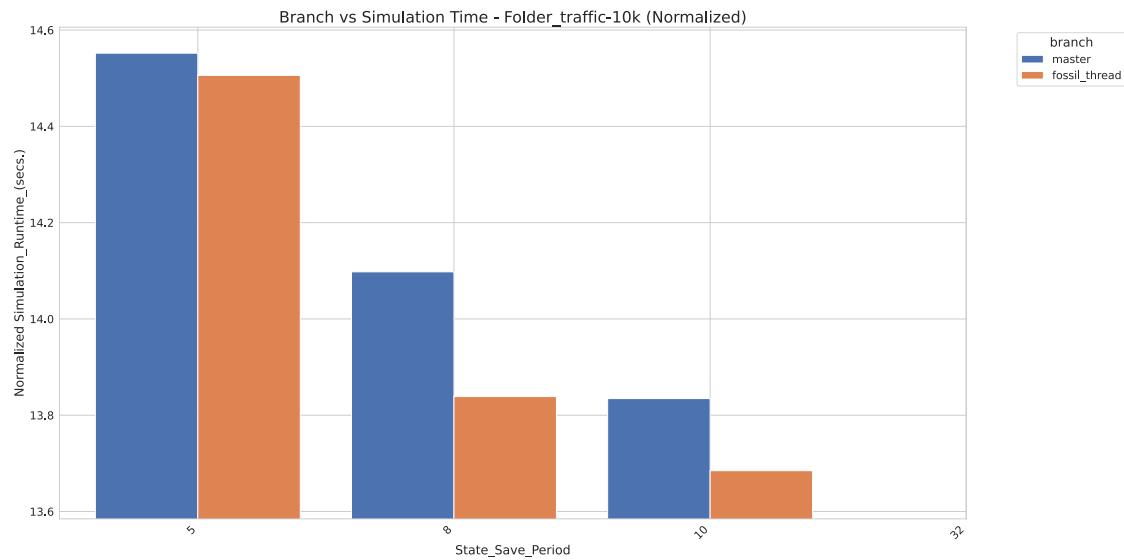


Figure 23: Branch vs Simulation Time_Folder_traffic-10k_normalized

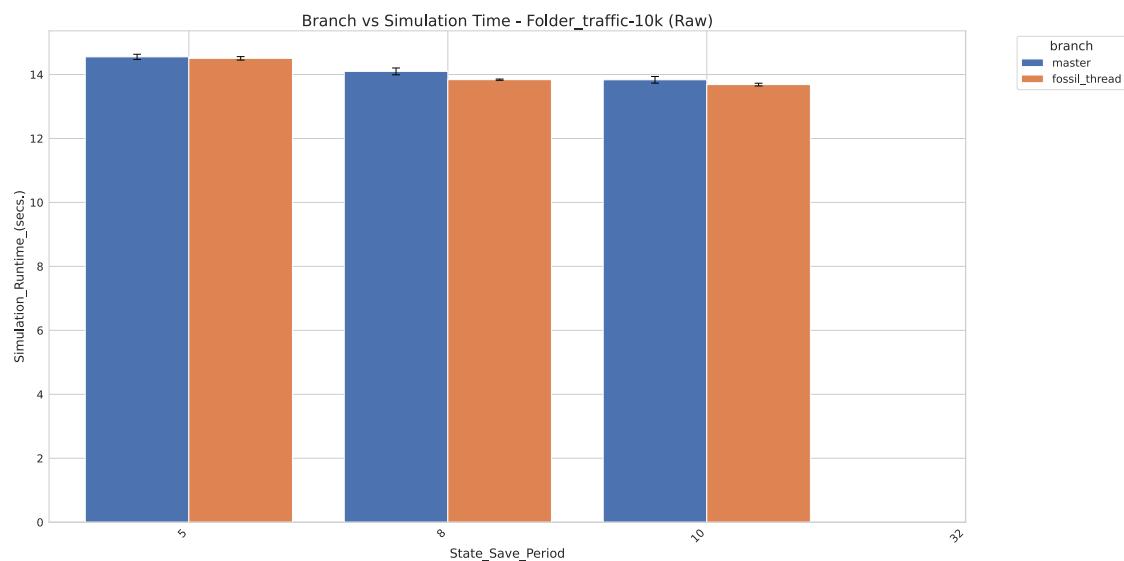


Figure 24: Branch vs Simulation Time_Folder_traffic-10k_raw

4 fossillocal

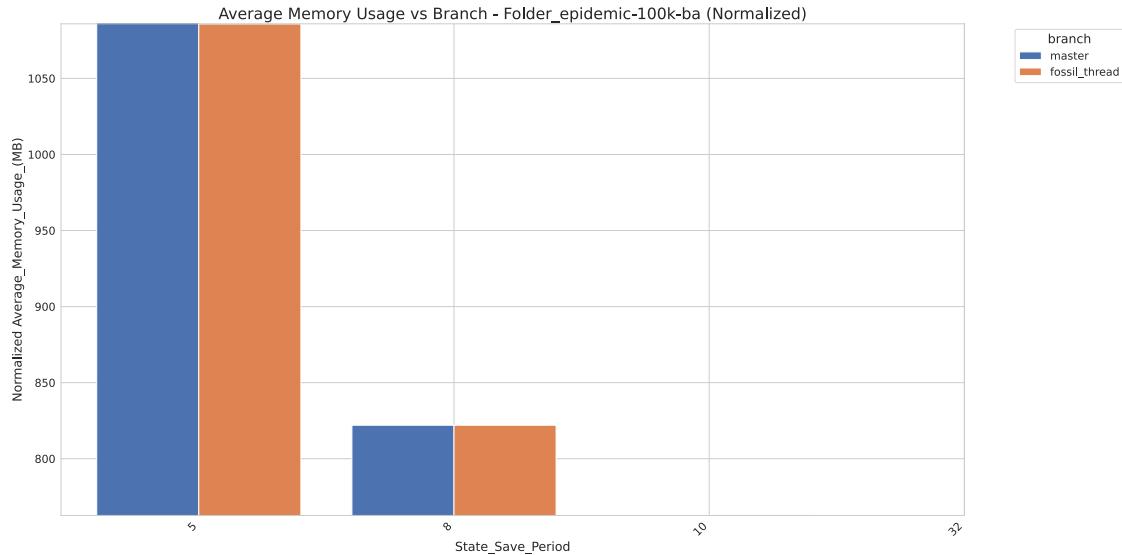


Figure 25: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

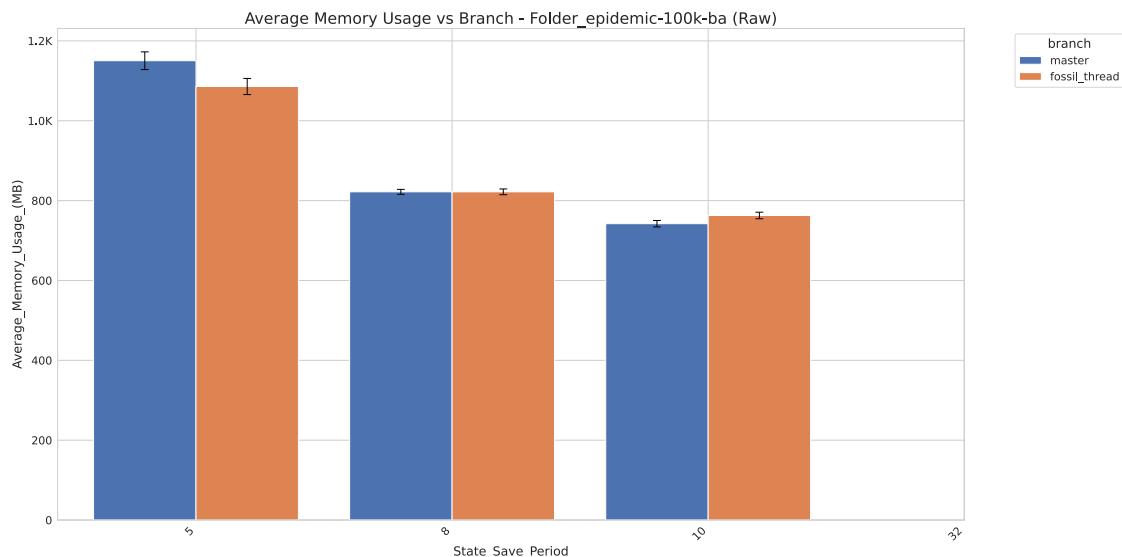


Figure 26: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

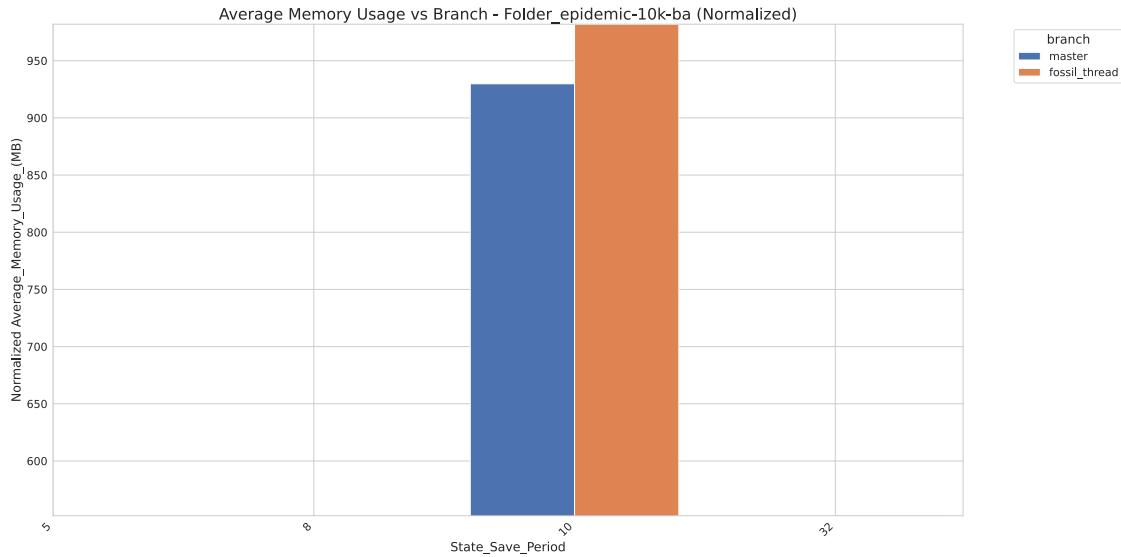


Figure 27: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

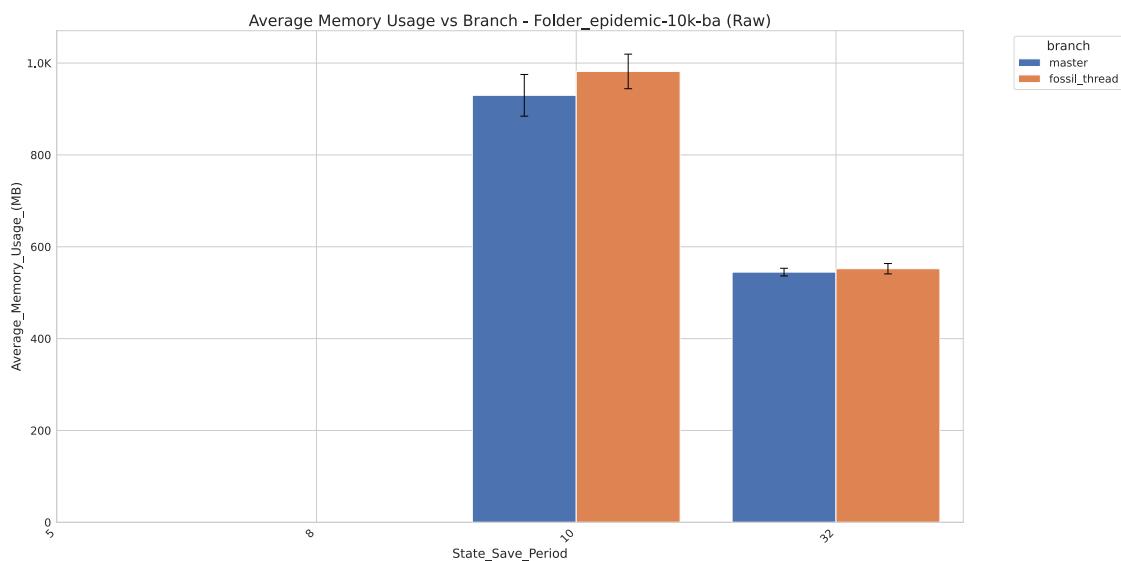


Figure 28: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

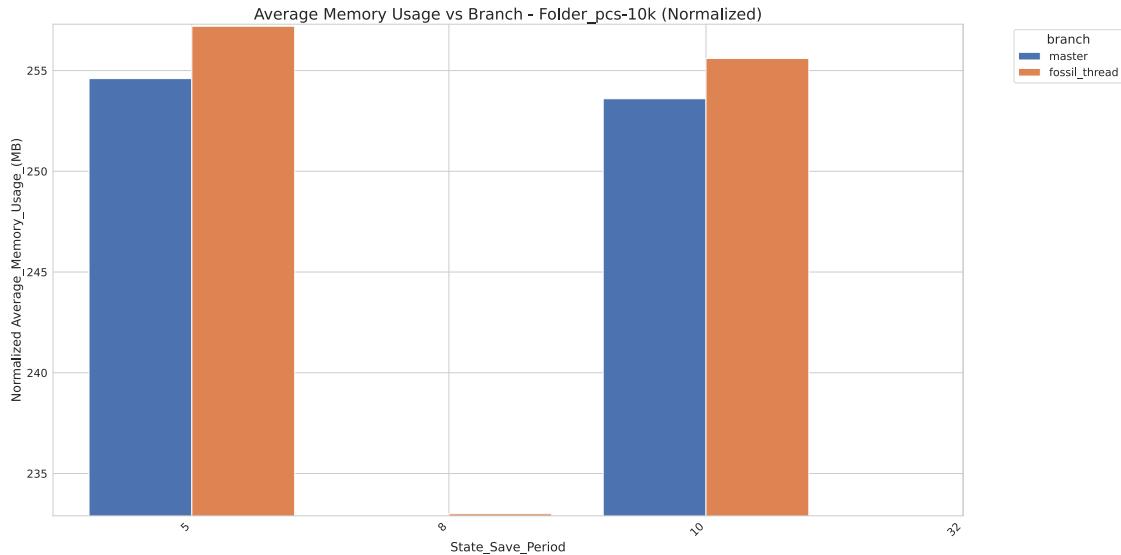


Figure 29: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

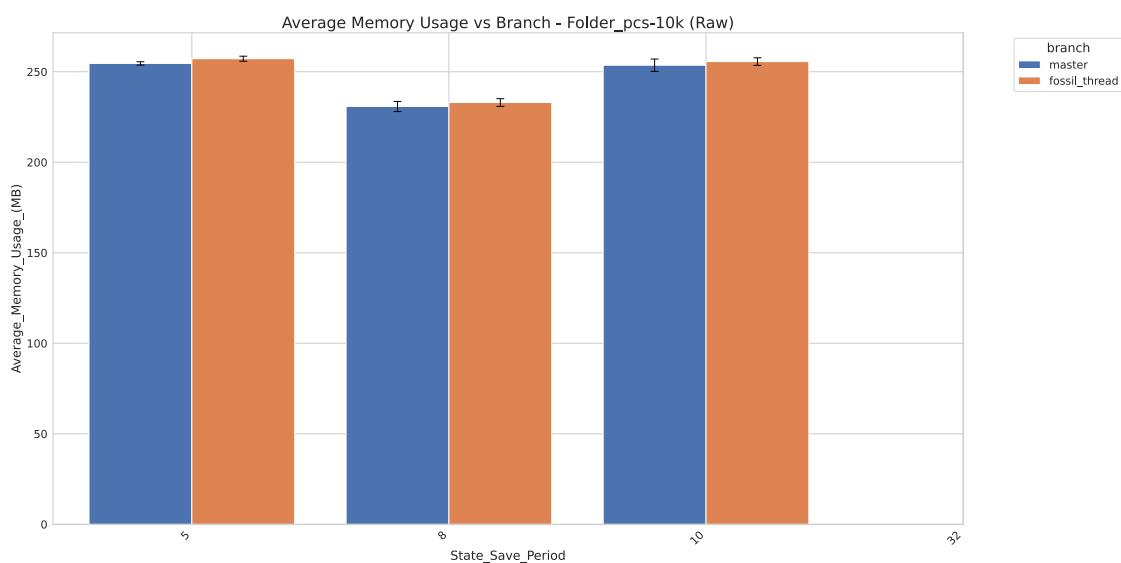


Figure 30: Average Memory Usage vs Branch_Folder_pcs-10k_raw

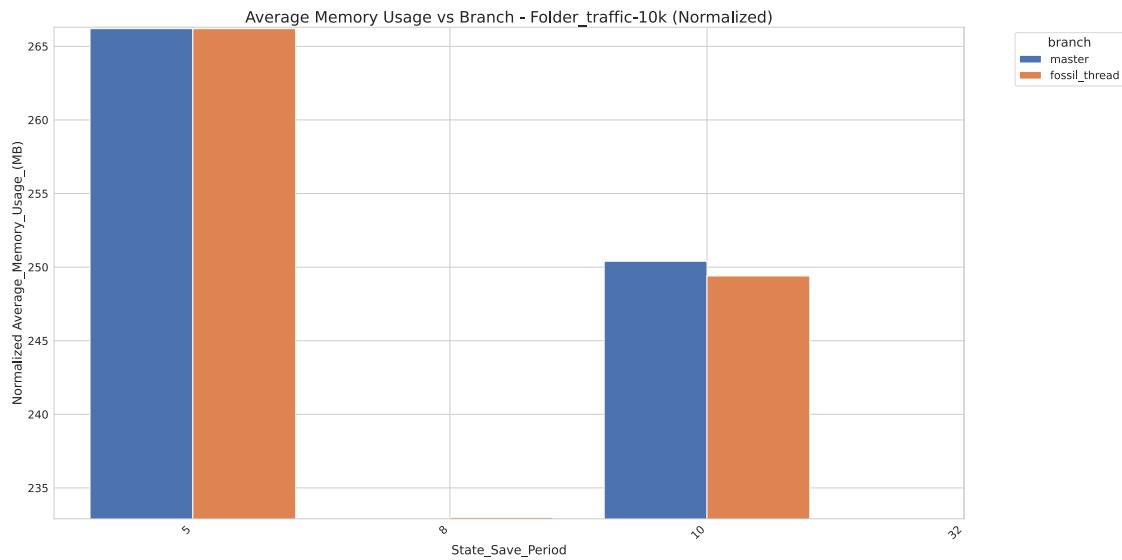


Figure 31: Average Memory Usage vs Branch_Folder_traffic-10k_normalized

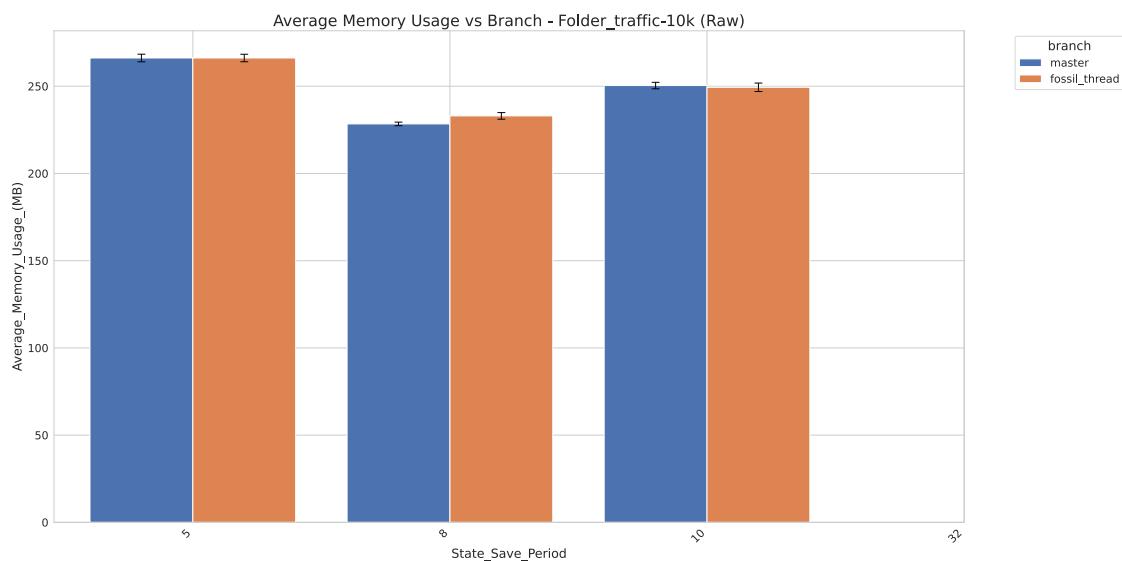


Figure 32: Average Memory Usage vs Branch_Folder_traffic-10k_raw

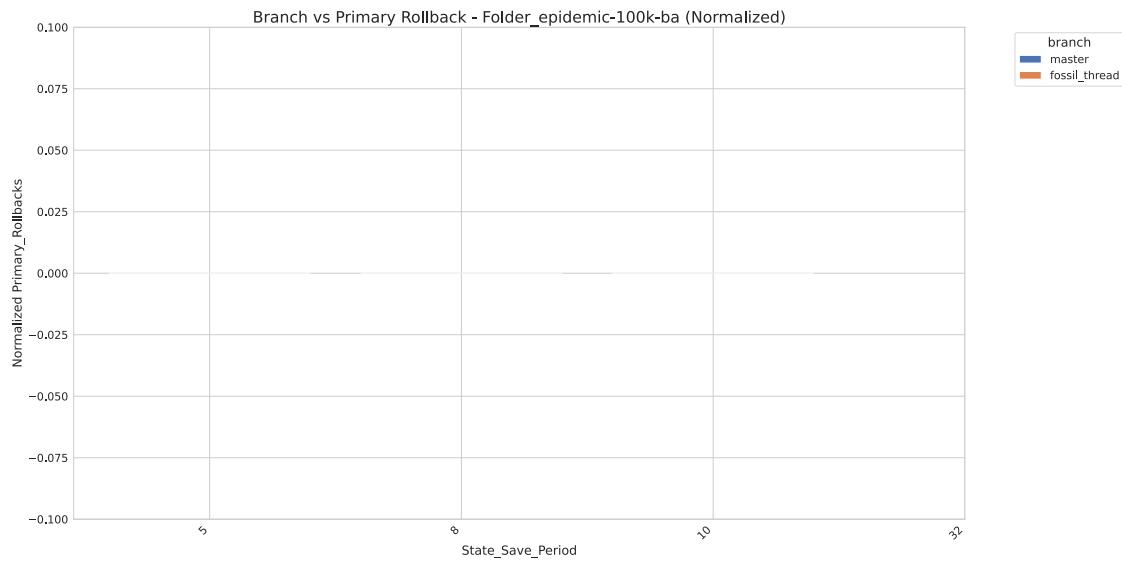


Figure 33: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

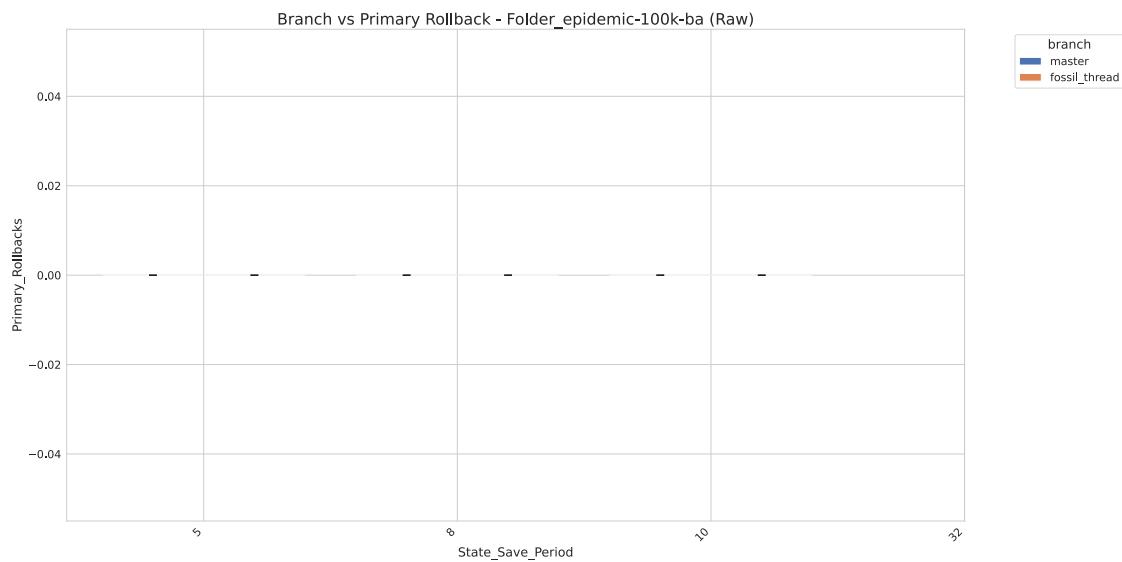


Figure 34: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

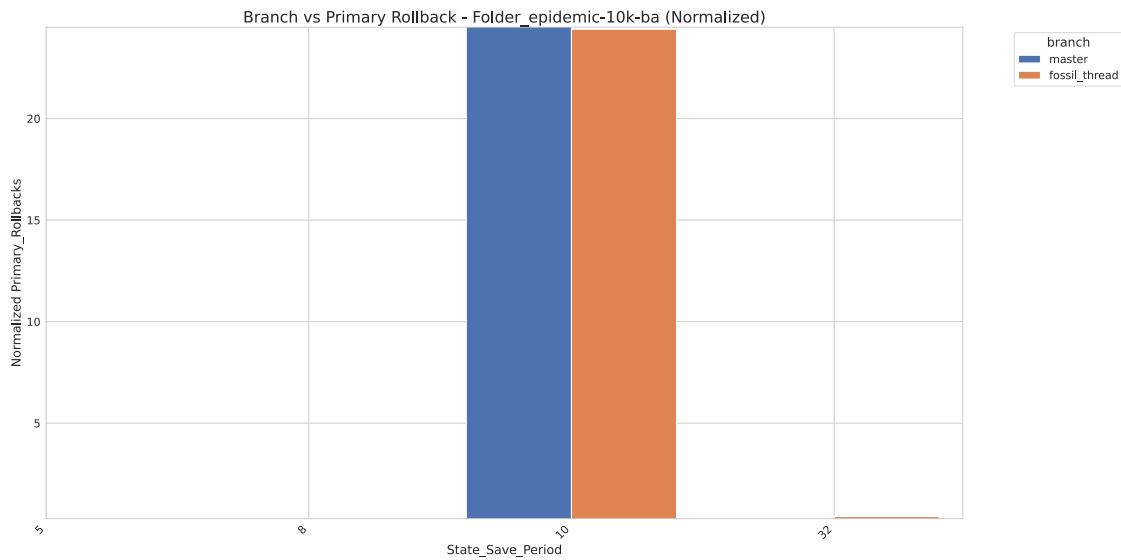


Figure 35: Branch vs Primary Rollback_Folder_epidemic-10k-ba_normalized

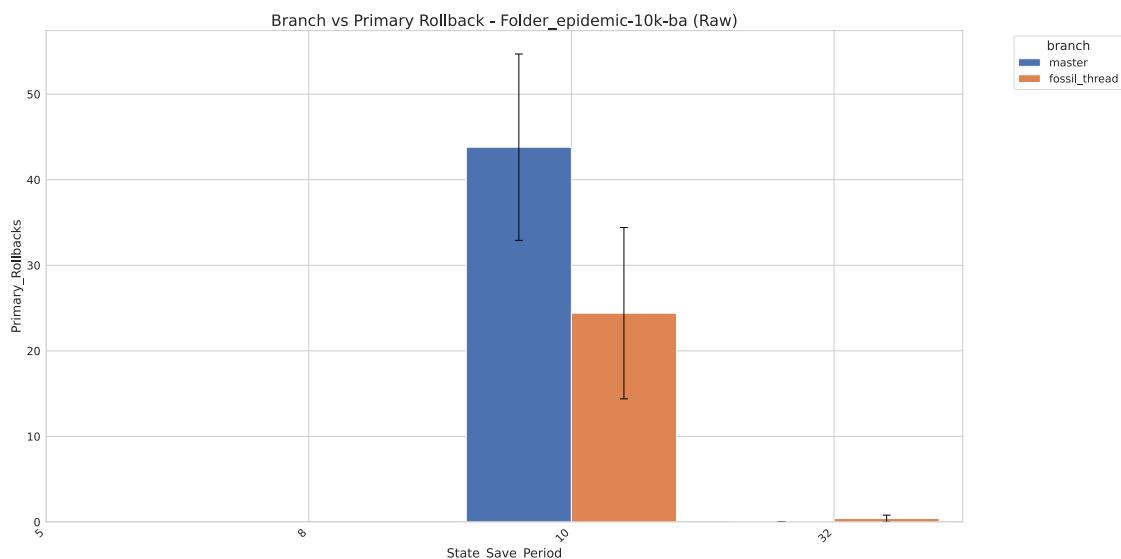


Figure 36: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

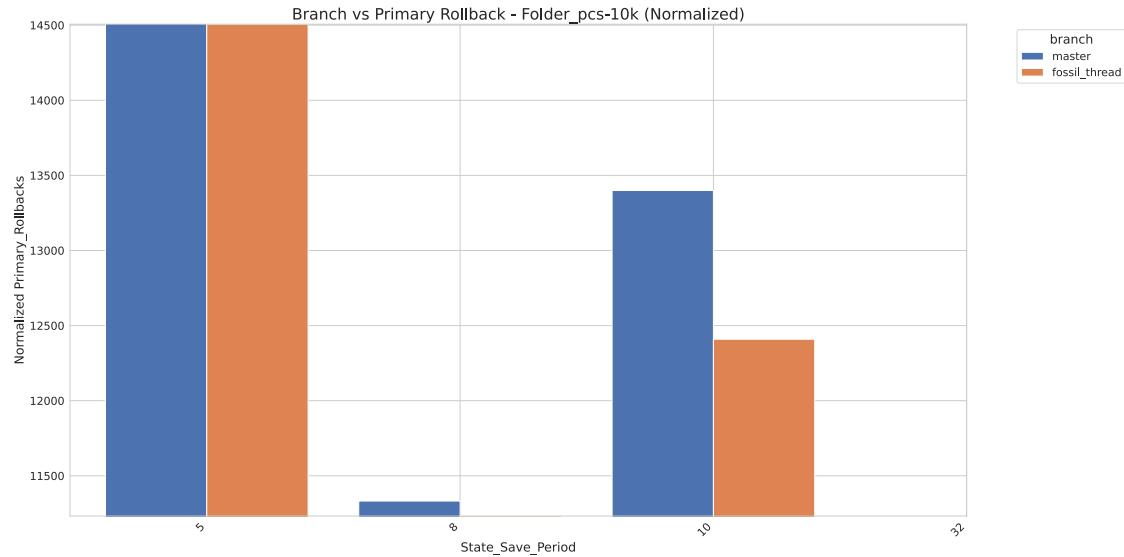


Figure 37: Branch vs Primary Rollback_Folder_pcs-10k_normalized

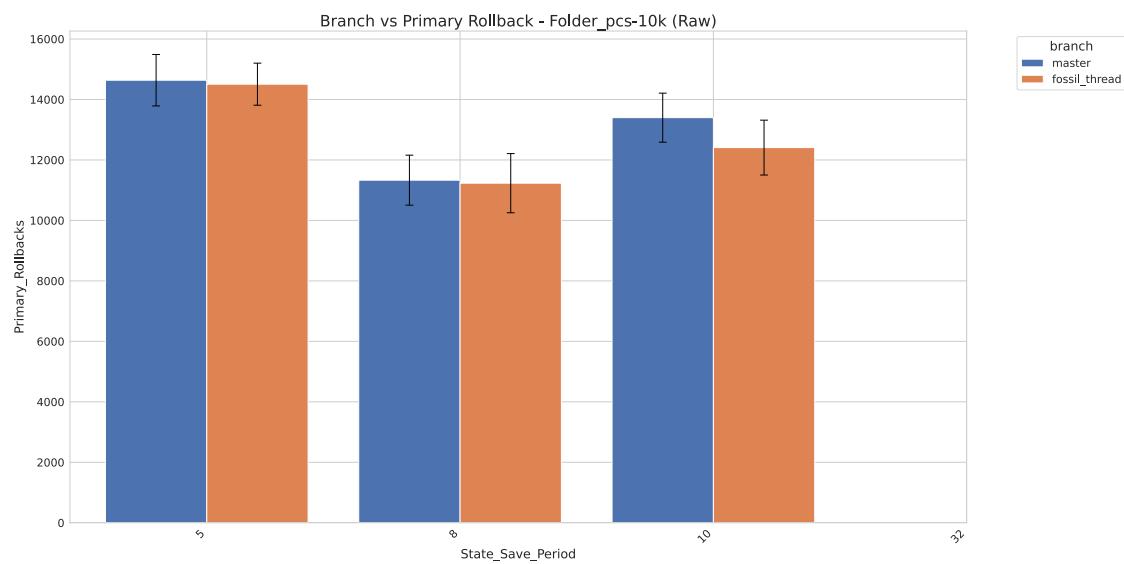


Figure 38: Branch vs Primary Rollback_Folder_pcs-10k_raw

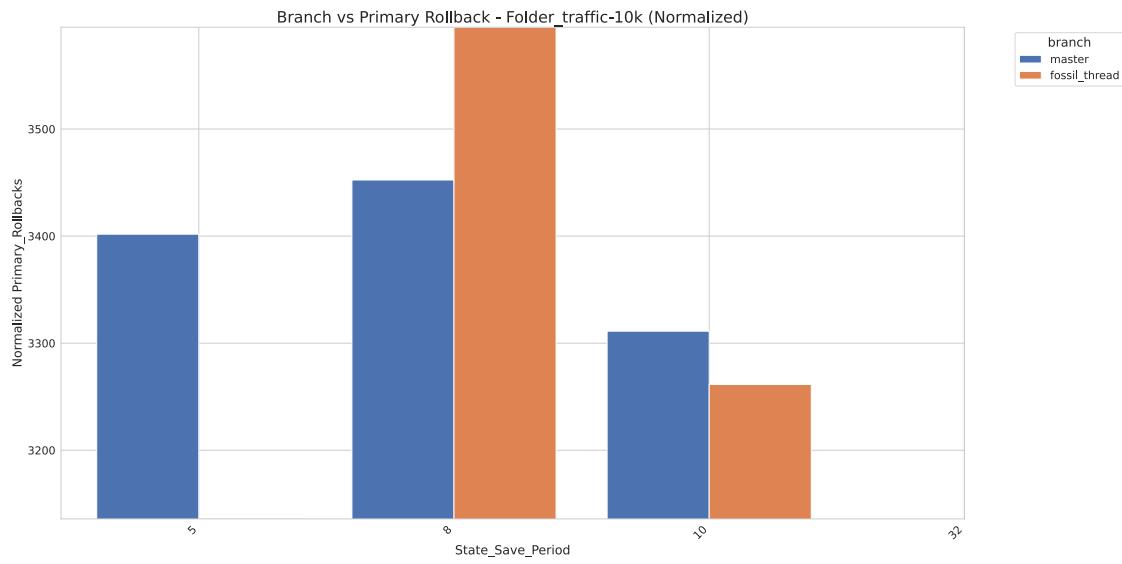


Figure 39: Branch vs Primary Rollback_Folder_traffic-10k_normalized

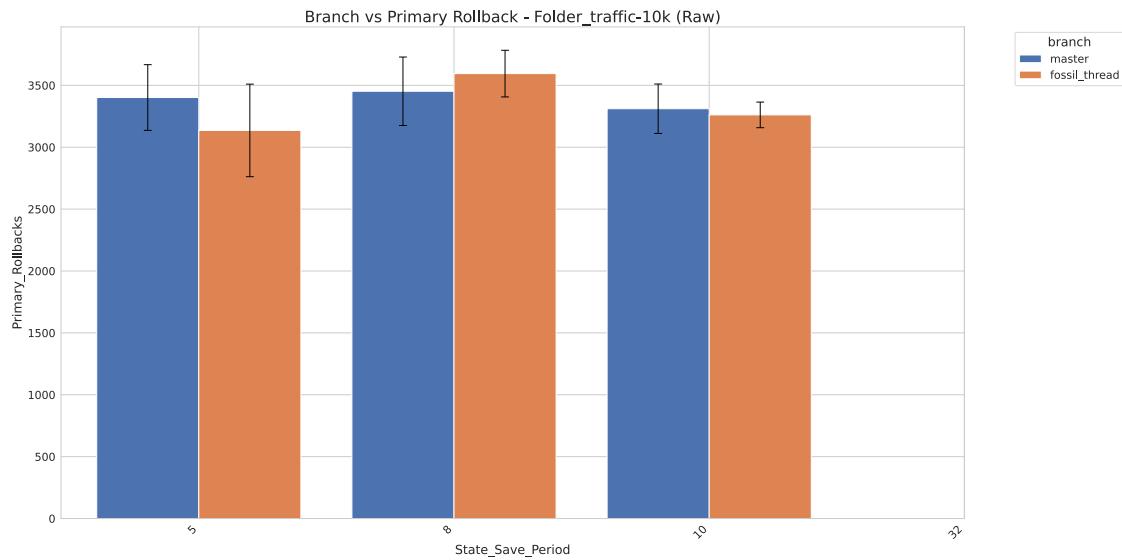


Figure 40: Branch vs Primary Rollback_Folder_traffic-10k_raw

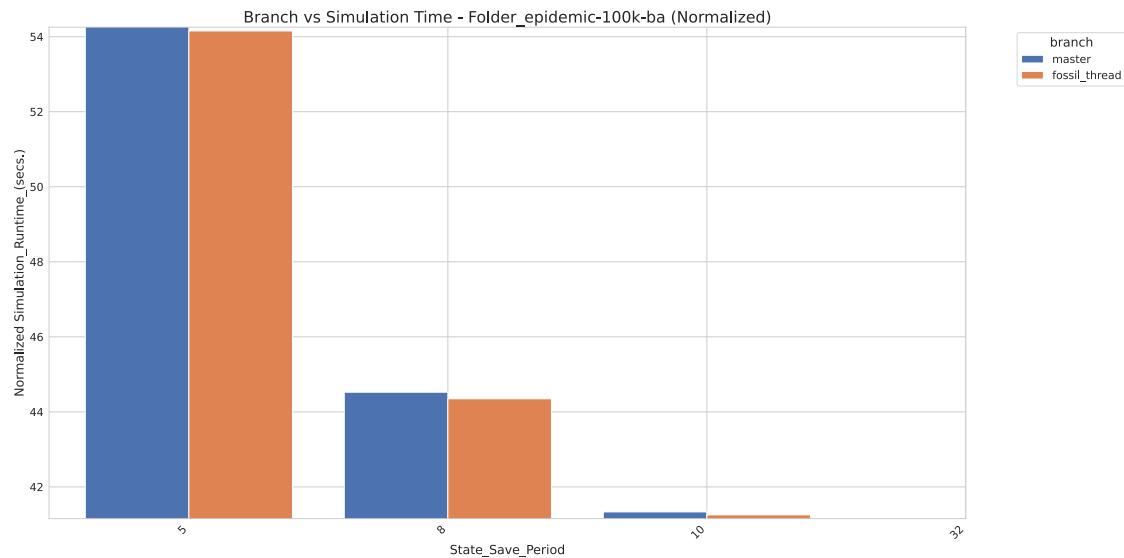


Figure 41: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

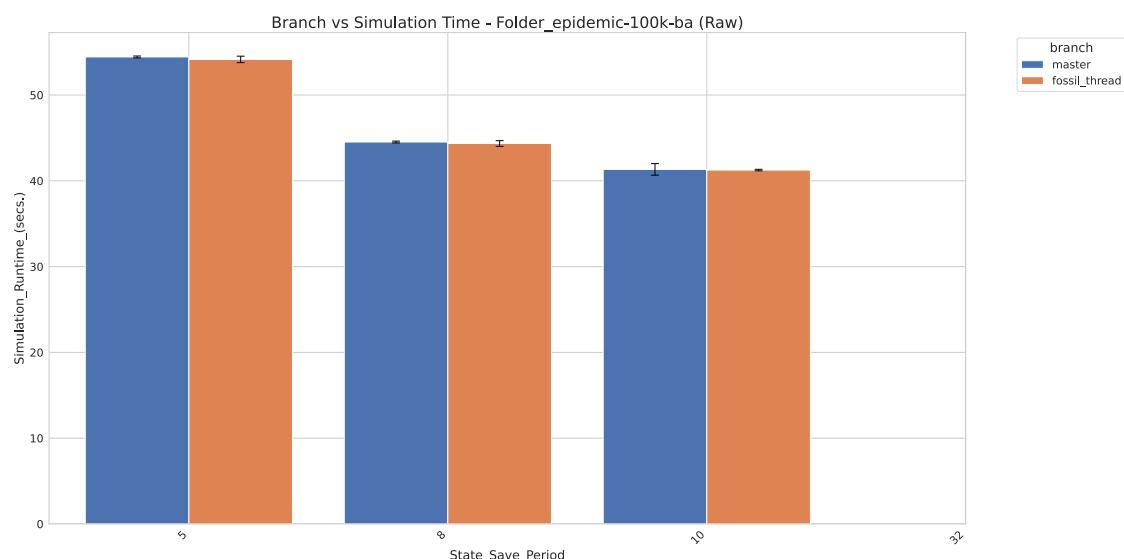


Figure 42: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

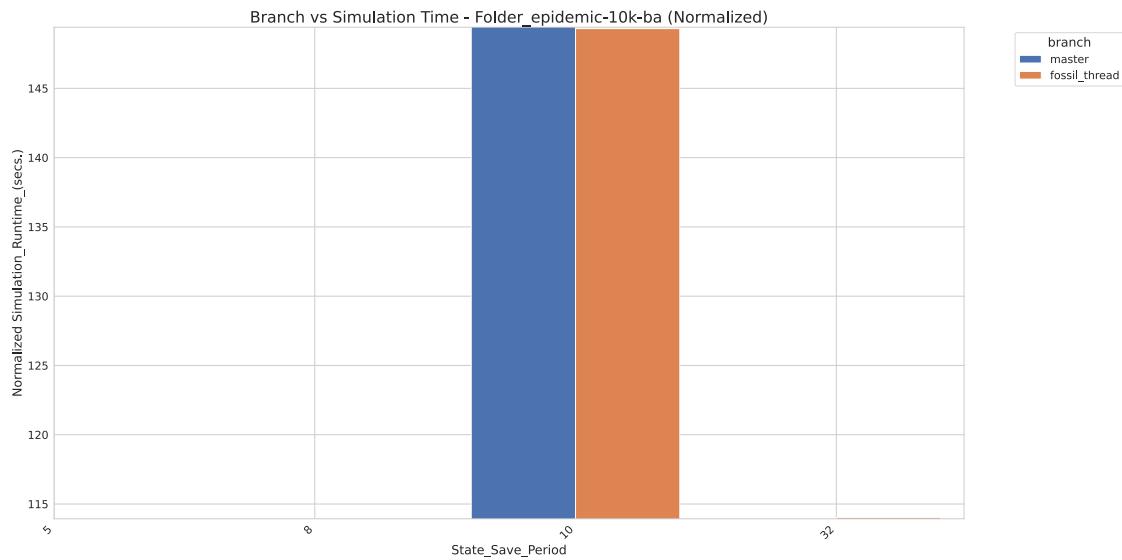


Figure 43: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

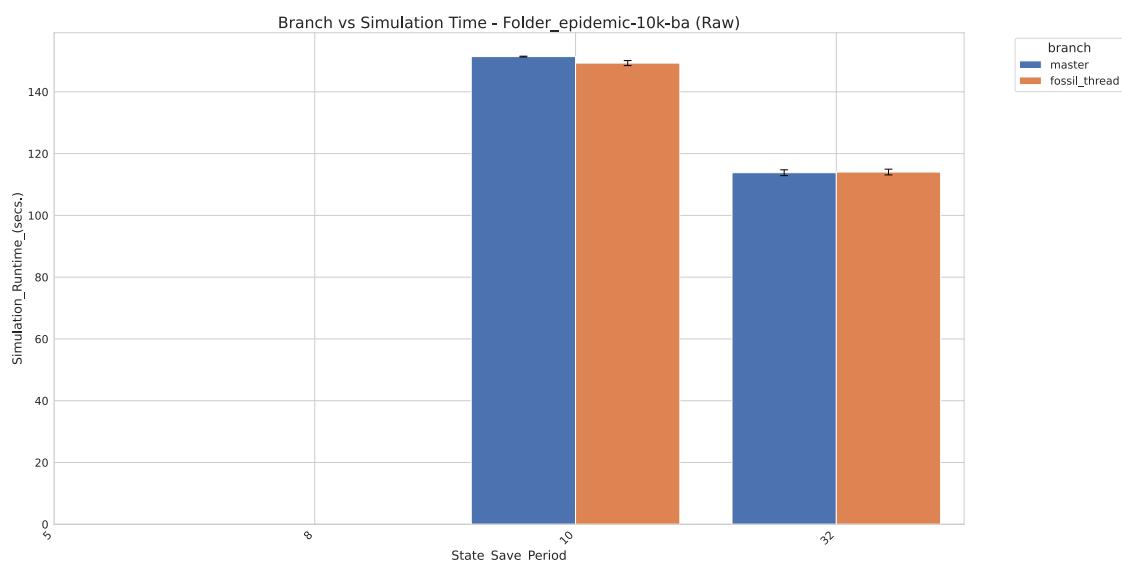


Figure 44: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

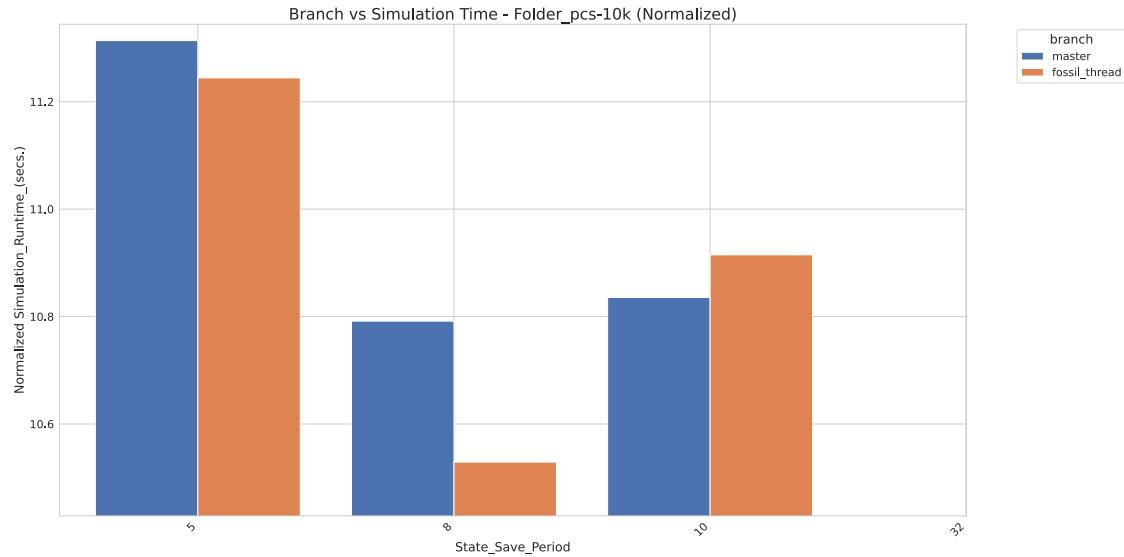


Figure 45: Branch vs Simulation Time_Folder_pcs-10k_normalized

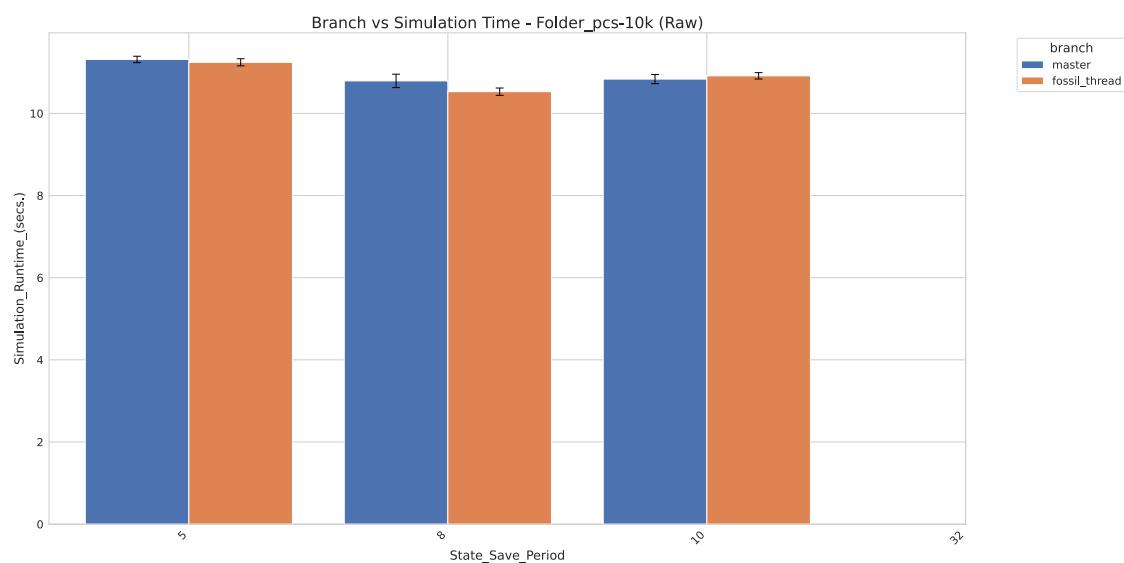


Figure 46: Branch vs Simulation Time_Folder_pcs-10k_raw

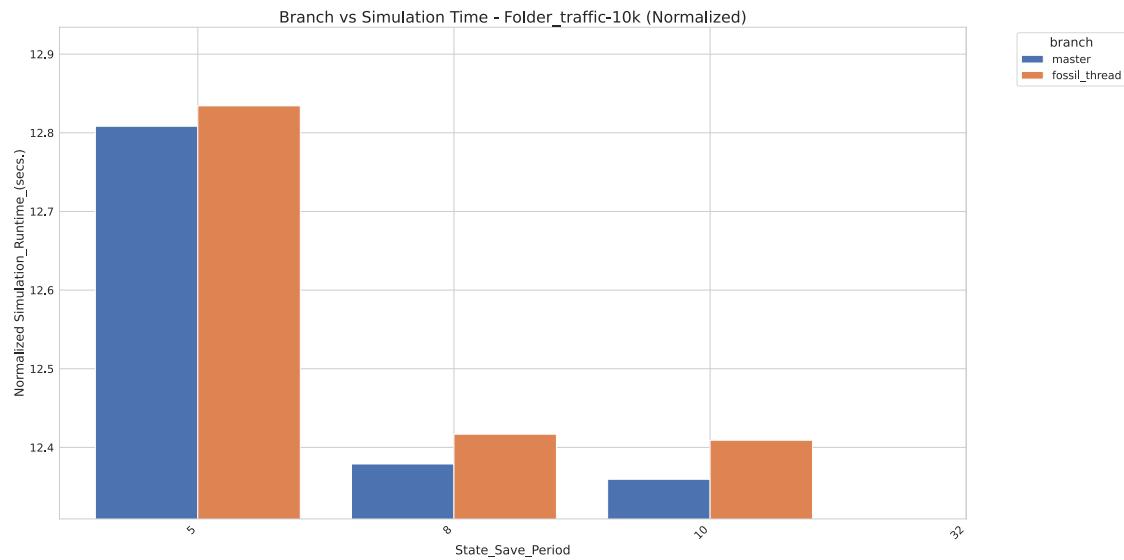


Figure 47: Branch vs Simulation Time_Folder_traffic-10k_normalized

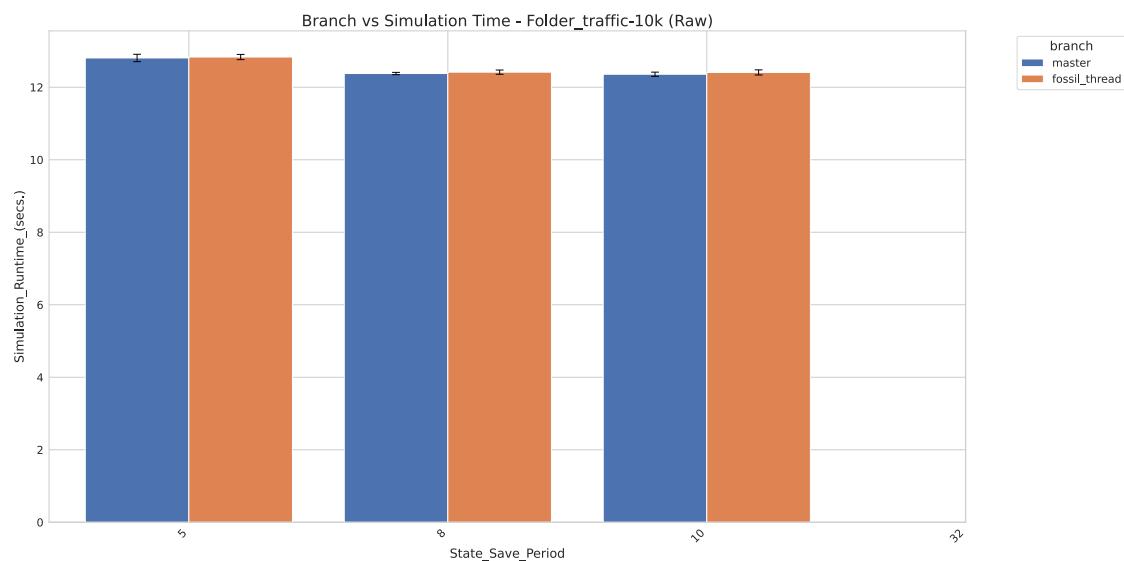


Figure 48: Branch vs Simulation Time_Folder_traffic-10k_raw

5 gaurav_unified

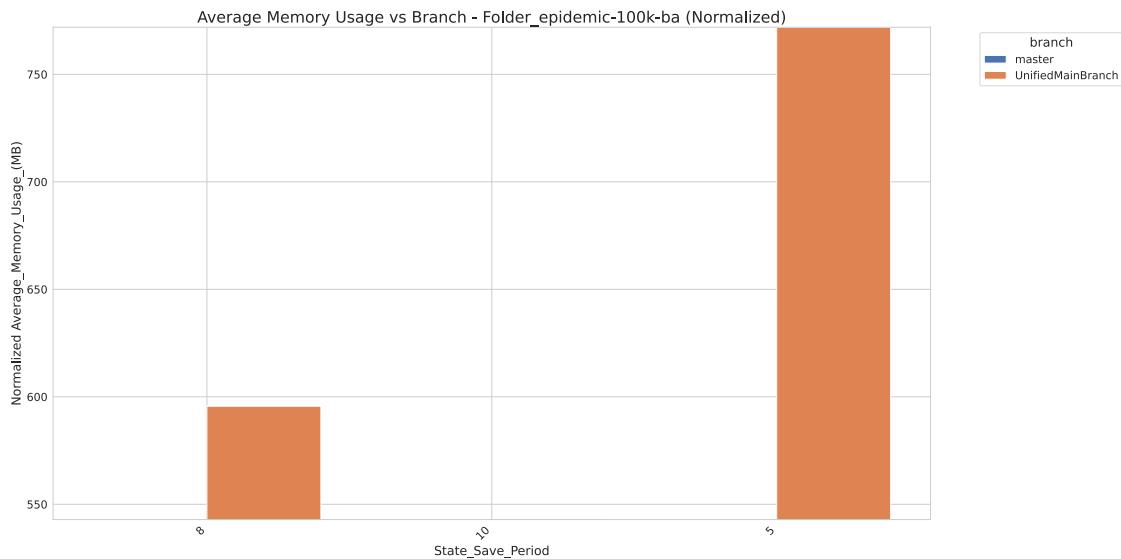


Figure 49: Average Memory Usage vs Branch.Folder.epidemic-100k-ba.normalized

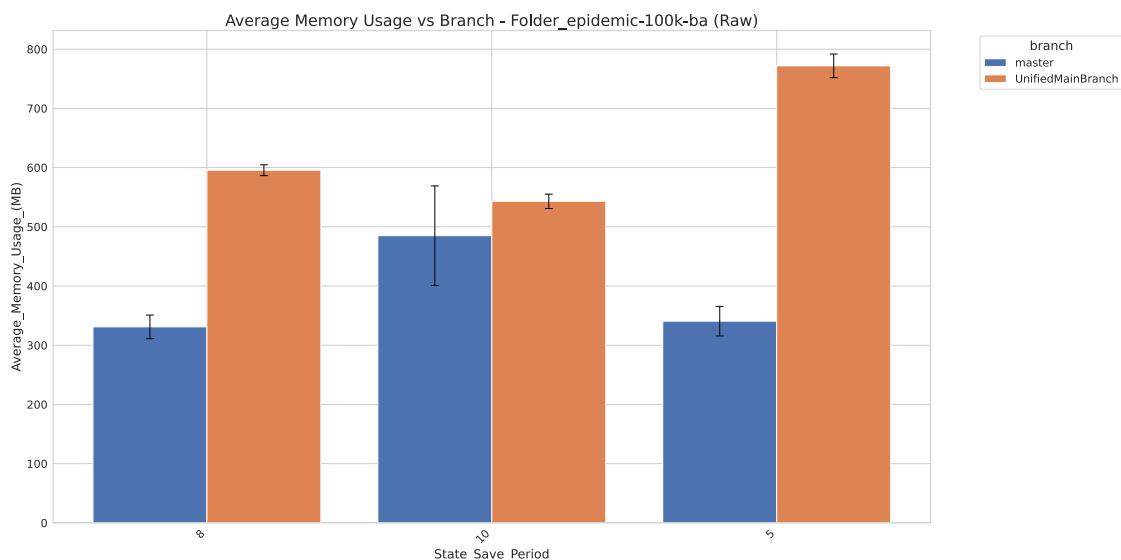


Figure 50: Average Memory Usage vs Branch.Folder.epidemic-100k-ba.raw

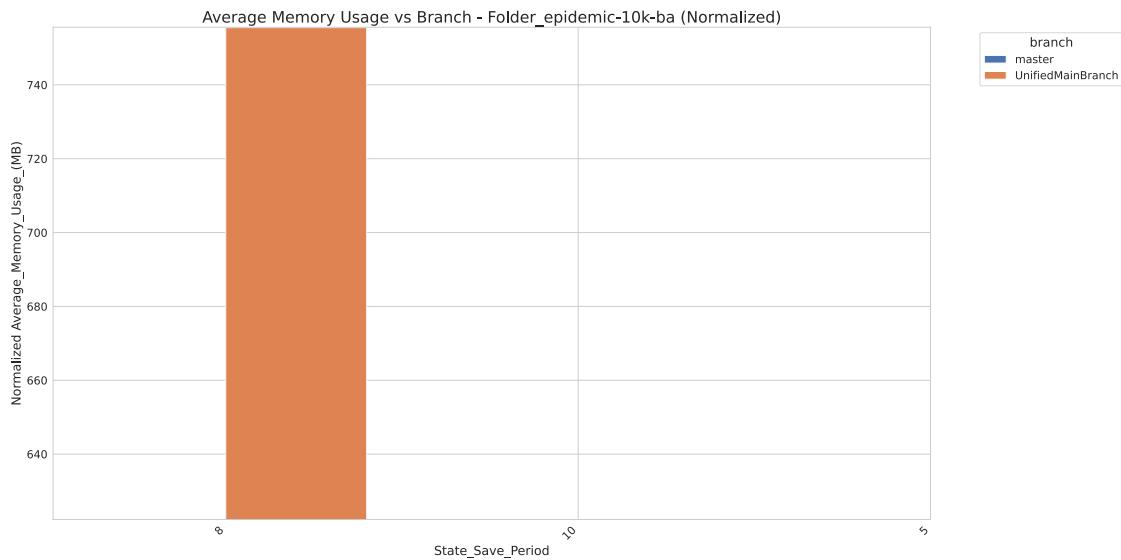


Figure 51: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

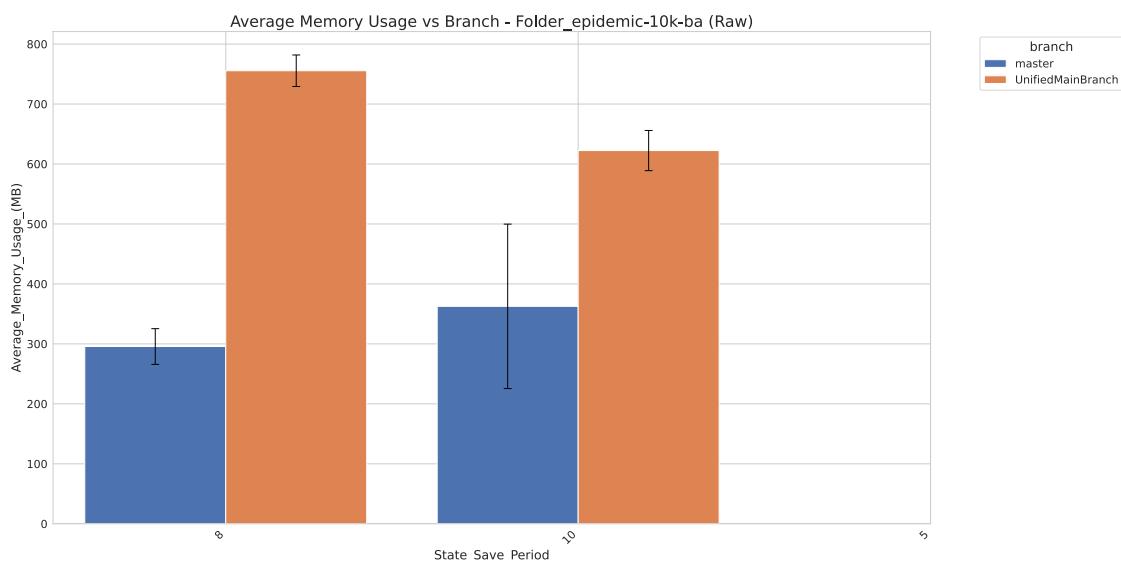


Figure 52: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

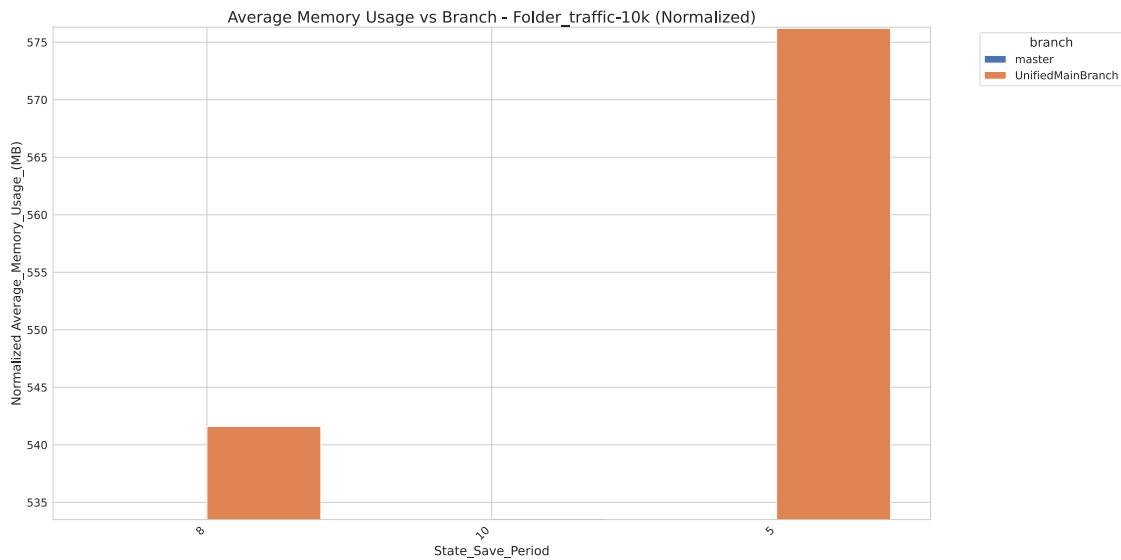


Figure 53: Average Memory Usage vs Branch_Folder_traffic-10k_normalized

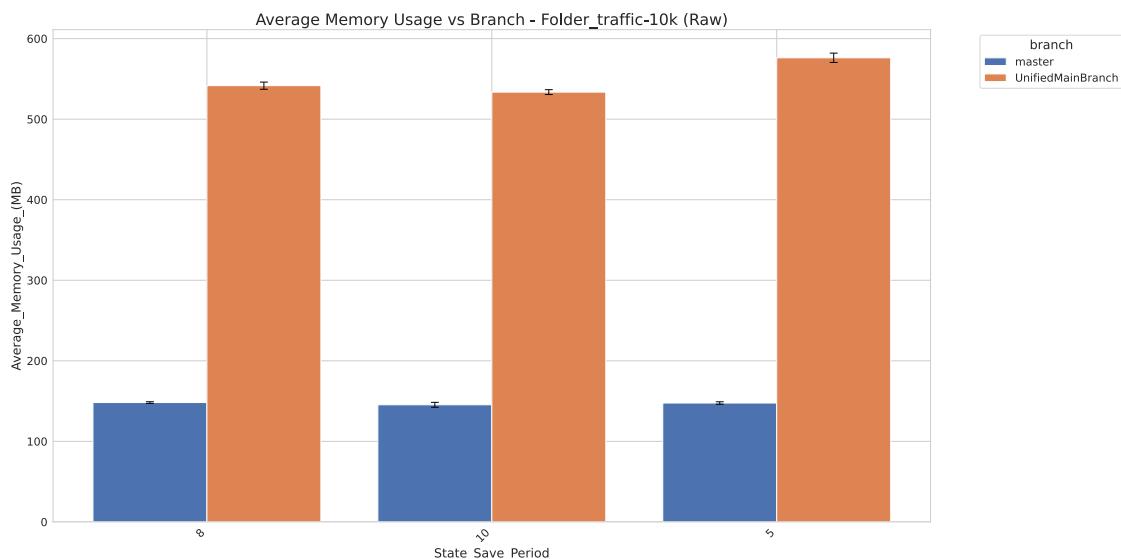


Figure 54: Average Memory Usage vs Branch_Folder_traffic-10k_raw

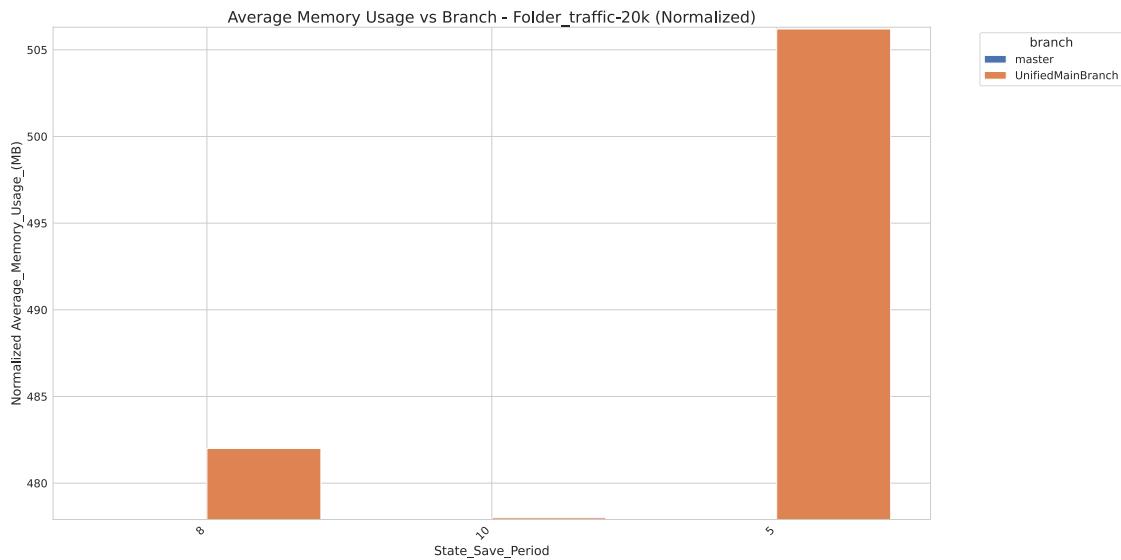


Figure 55: Average Memory Usage vs Branch_Folder_traffic-20k_normalized

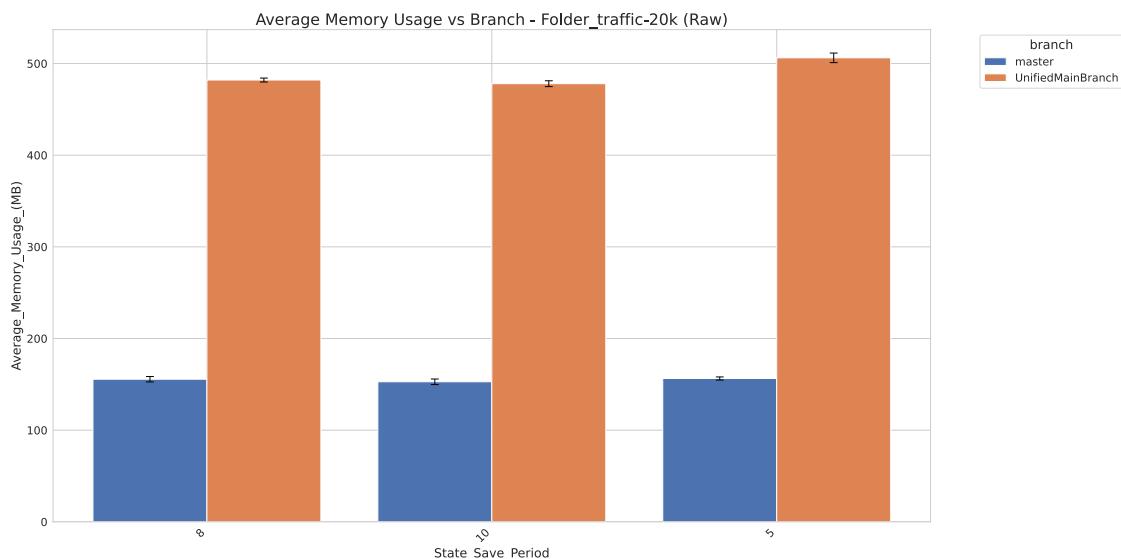


Figure 56: Average Memory Usage vs Branch_Folder_traffic-20k_raw

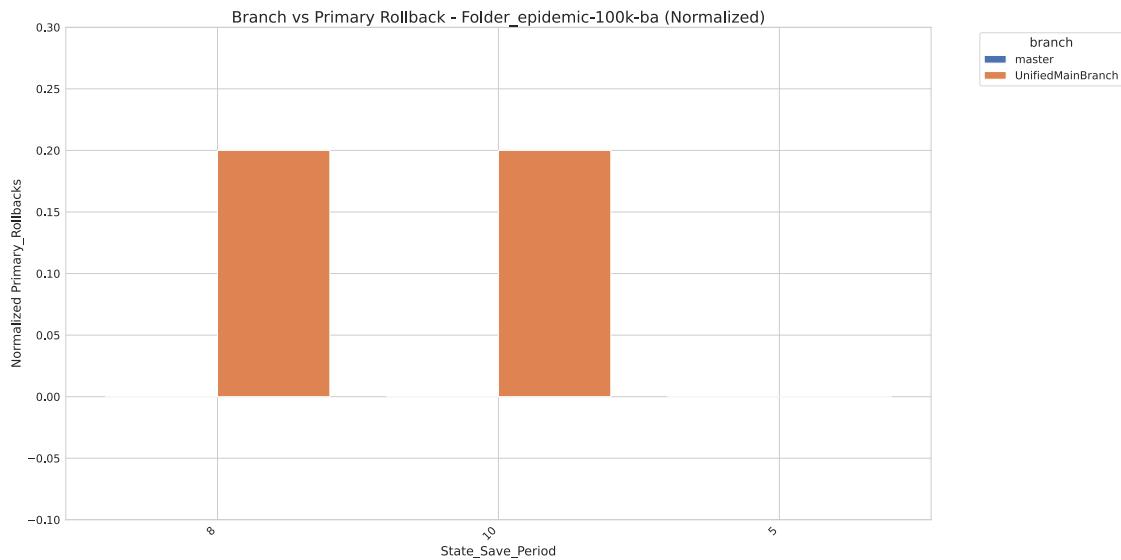


Figure 57: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

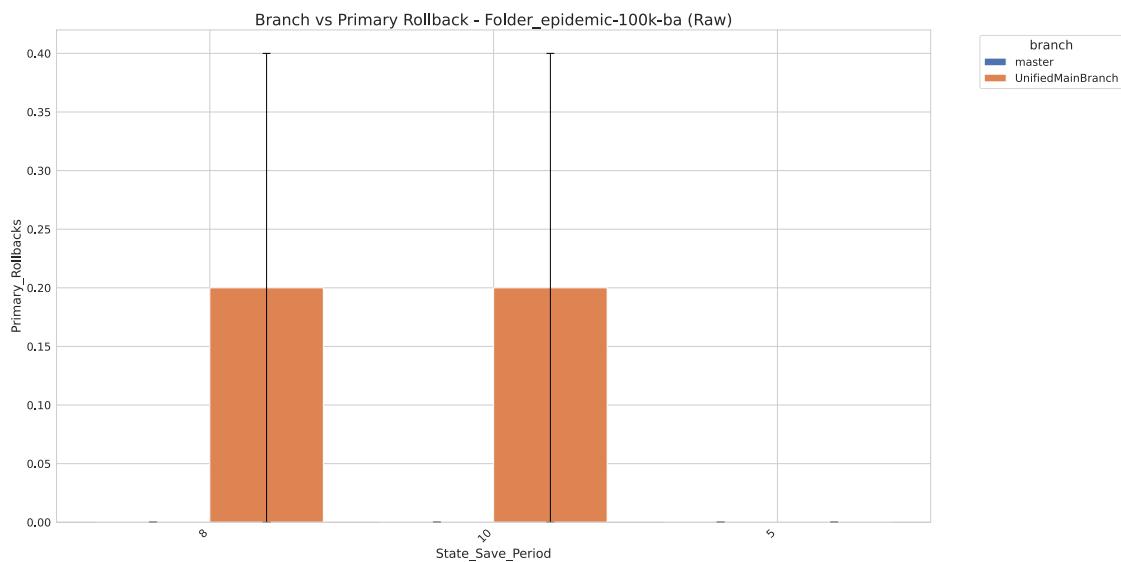


Figure 58: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

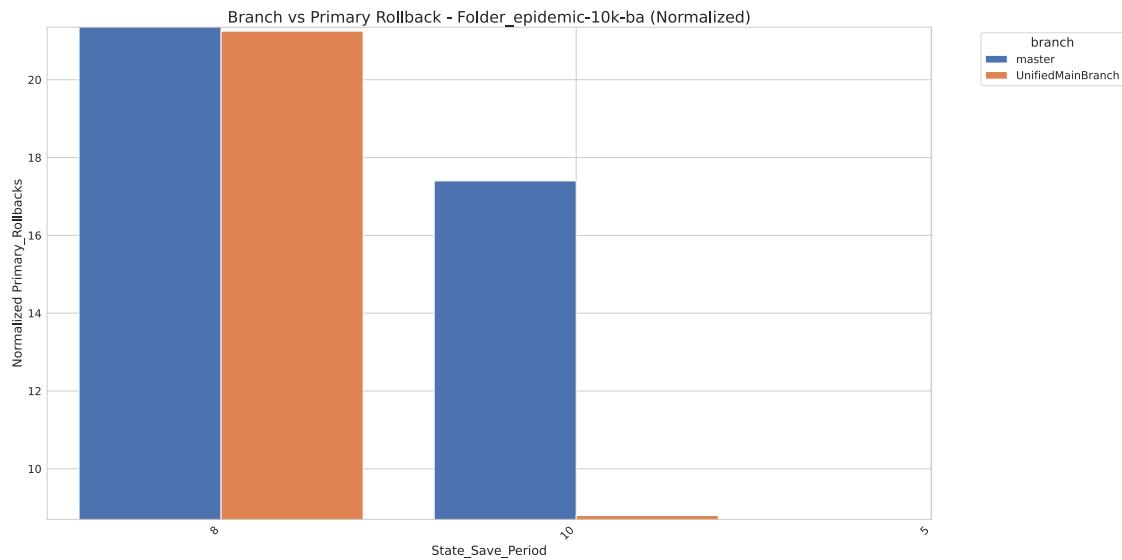


Figure 59: Branch vs Primary Rollback_Folder_epidemic-10k-ba_normalized

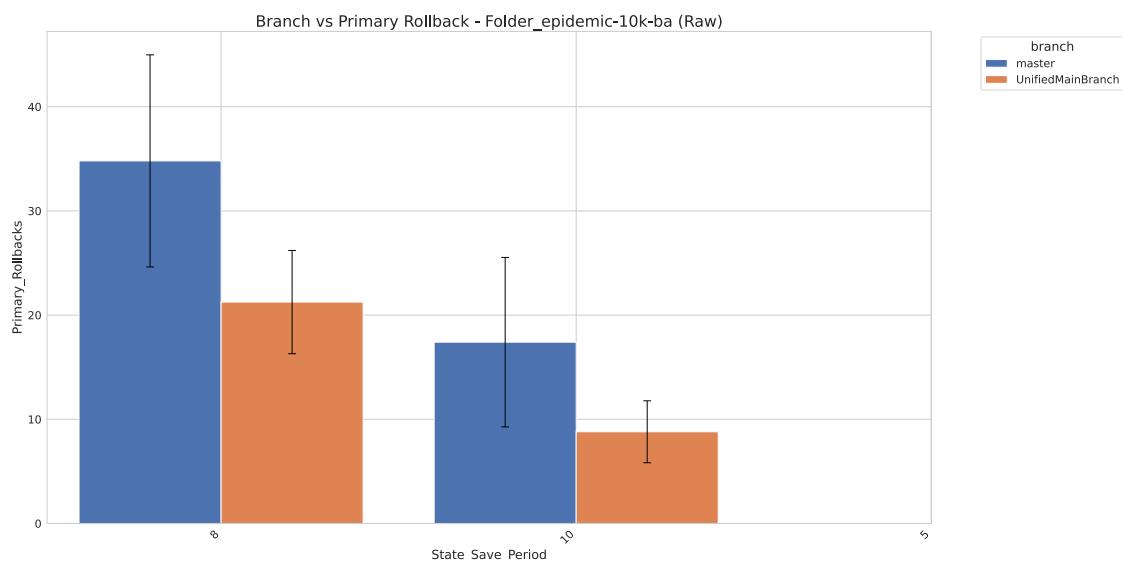


Figure 60: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

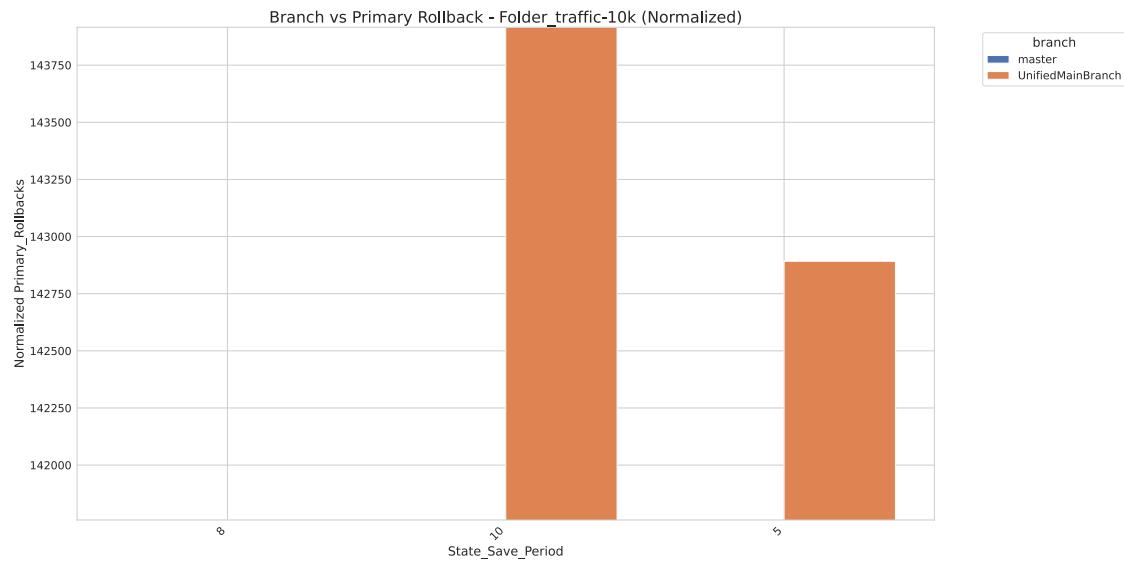


Figure 61: Branch vs Primary Rollback_Folder_traffic-10k_normalized

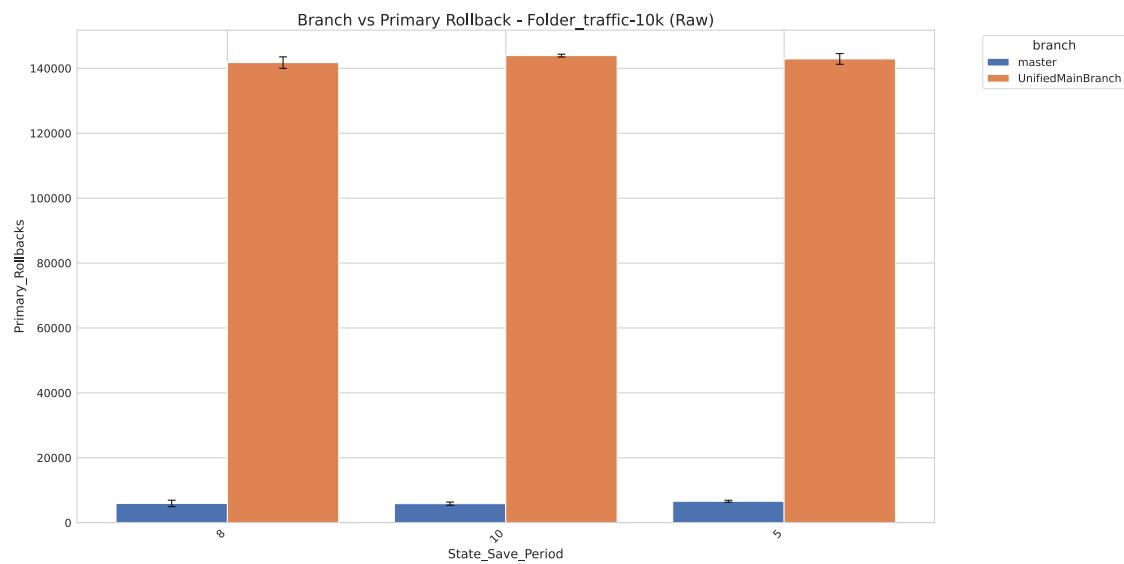


Figure 62: Branch vs Primary Rollback_Folder_traffic-10k_raw

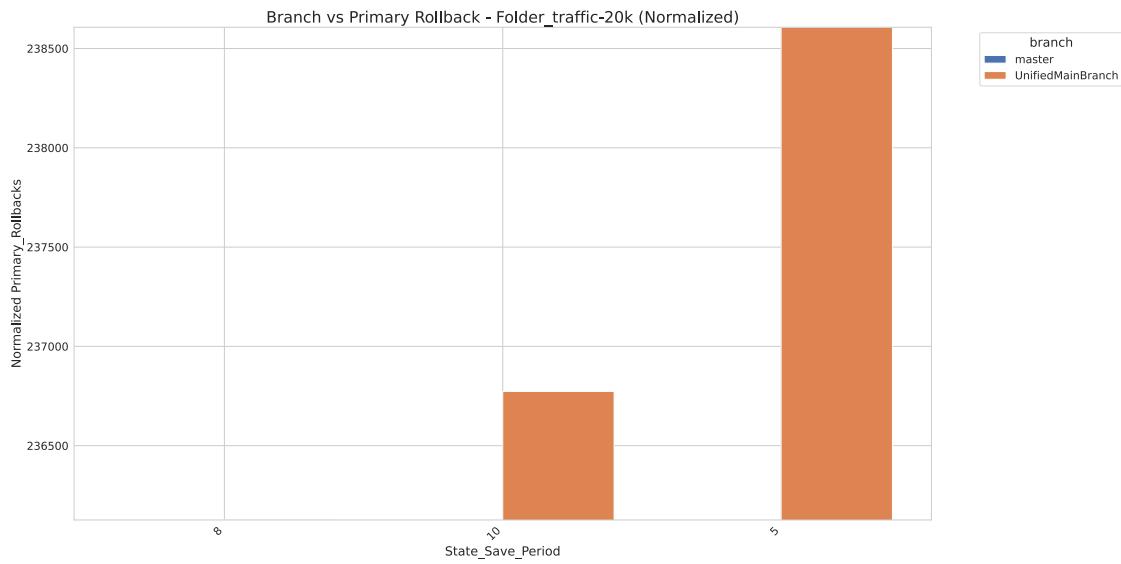


Figure 63: Branch vs Primary Rollback_Folder_traffic-20k_normalized

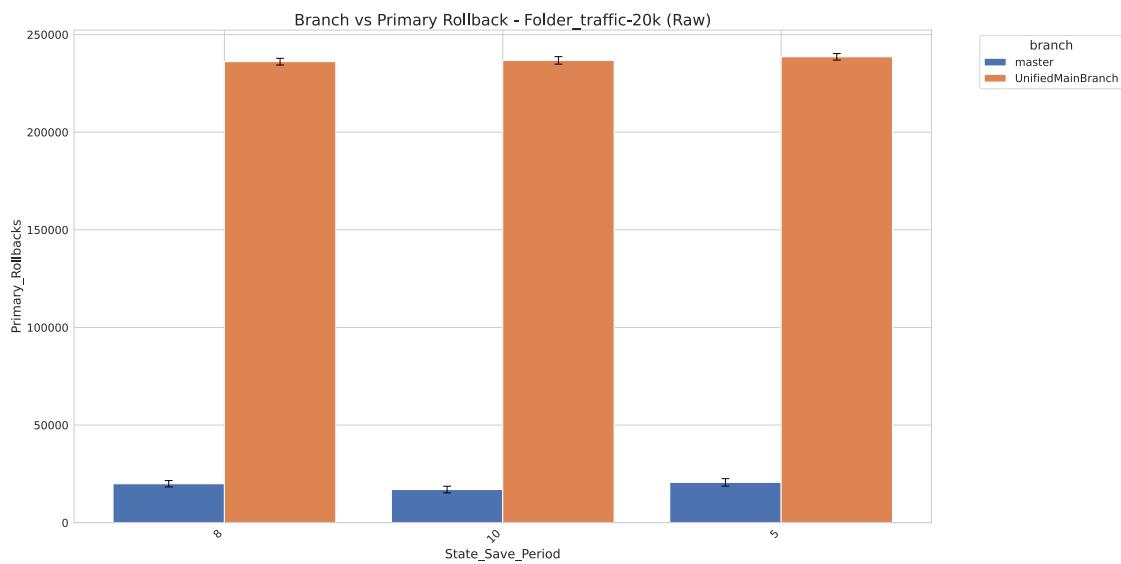


Figure 64: Branch vs Primary Rollback_Folder_traffic-20k_raw

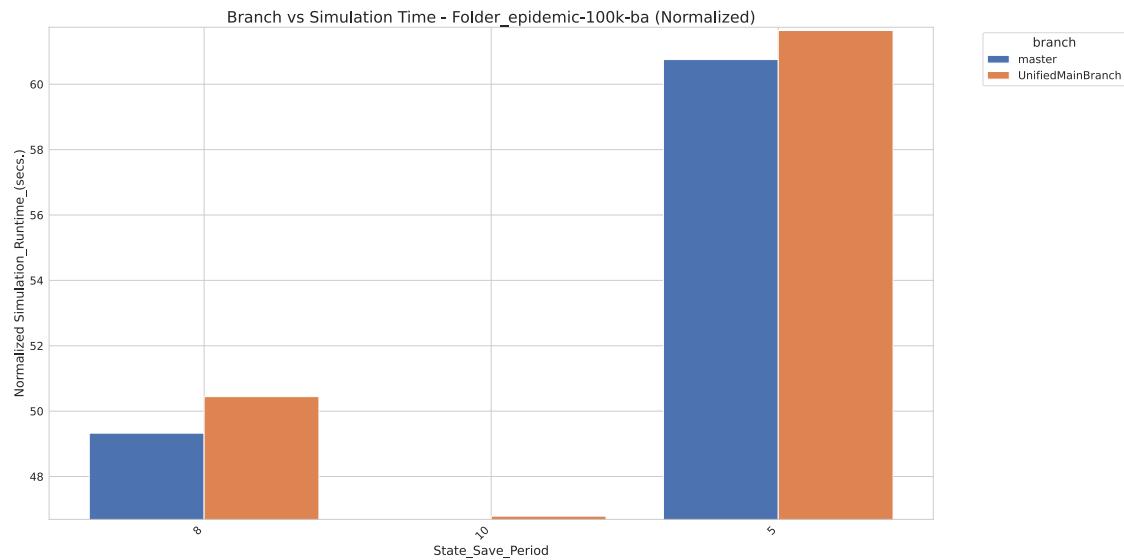


Figure 65: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

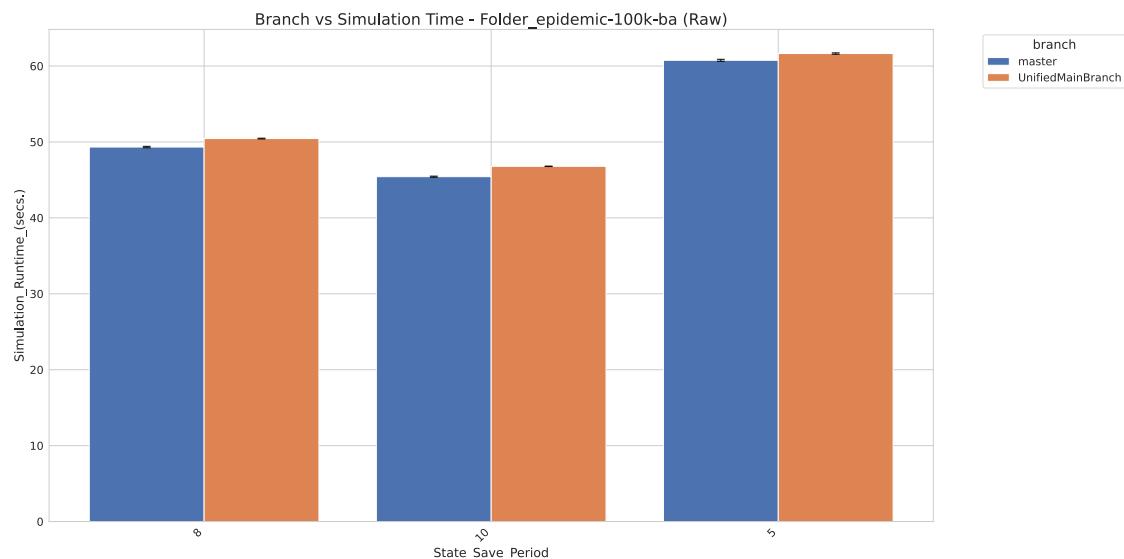


Figure 66: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

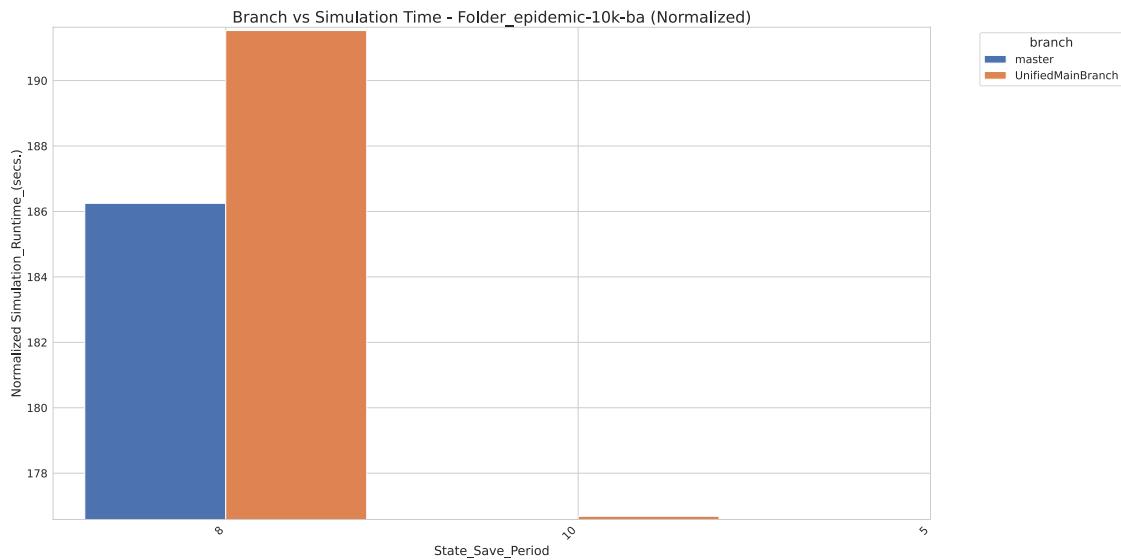


Figure 67: Branch vs Simulation Time_Folder_e pidemic-10k-ba_normalized

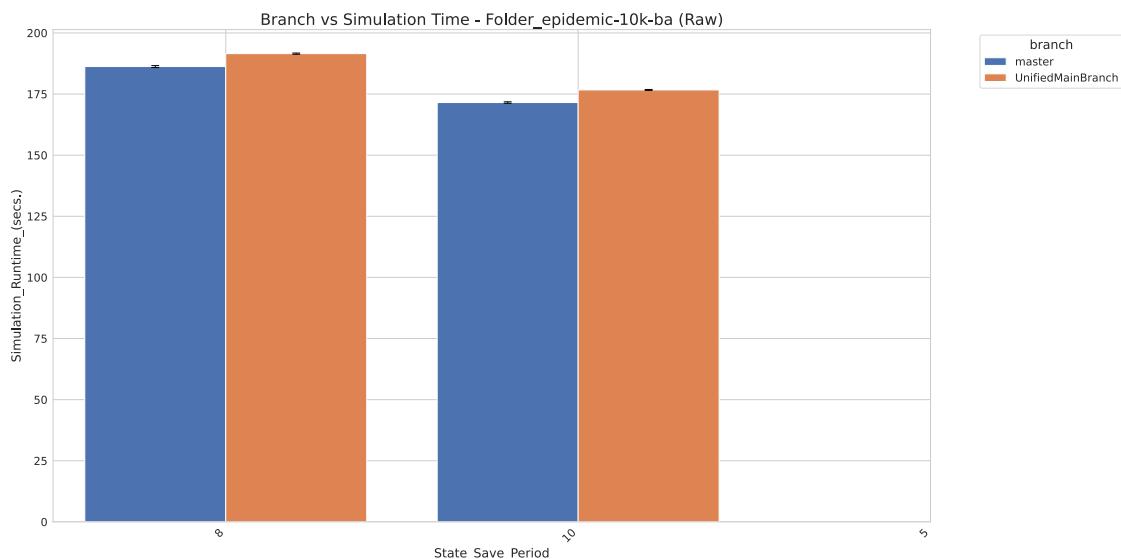


Figure 68: Branch vs Simulation Time_Folder_e pidemic-10k-ba_raw

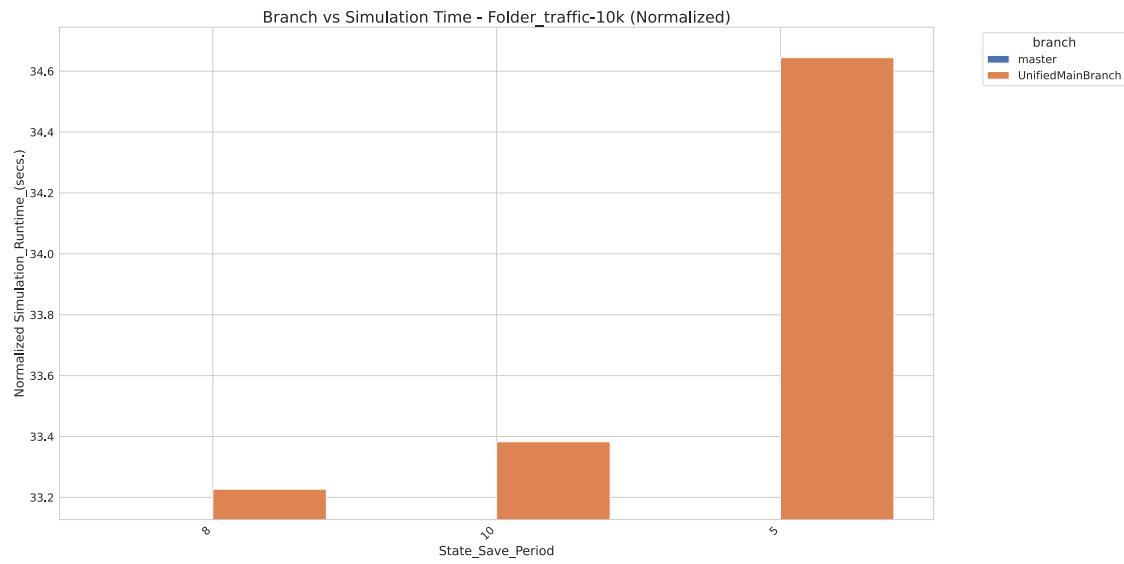


Figure 69: Branch vs Simulation Time_Folder_traffic-10k_normalized

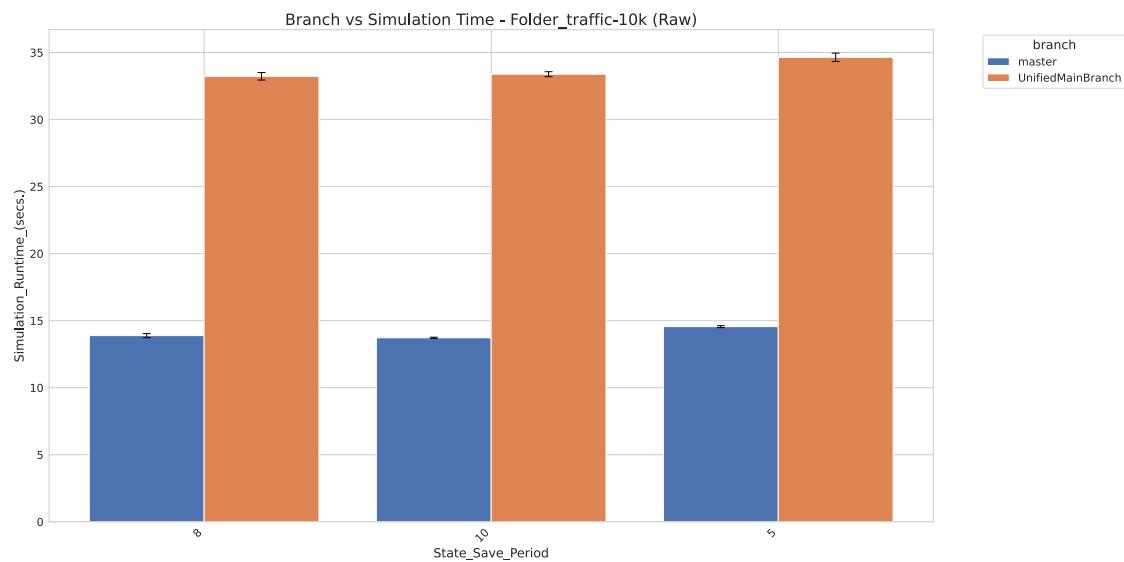


Figure 70: Branch vs Simulation Time_Folder_traffic-10k_raw

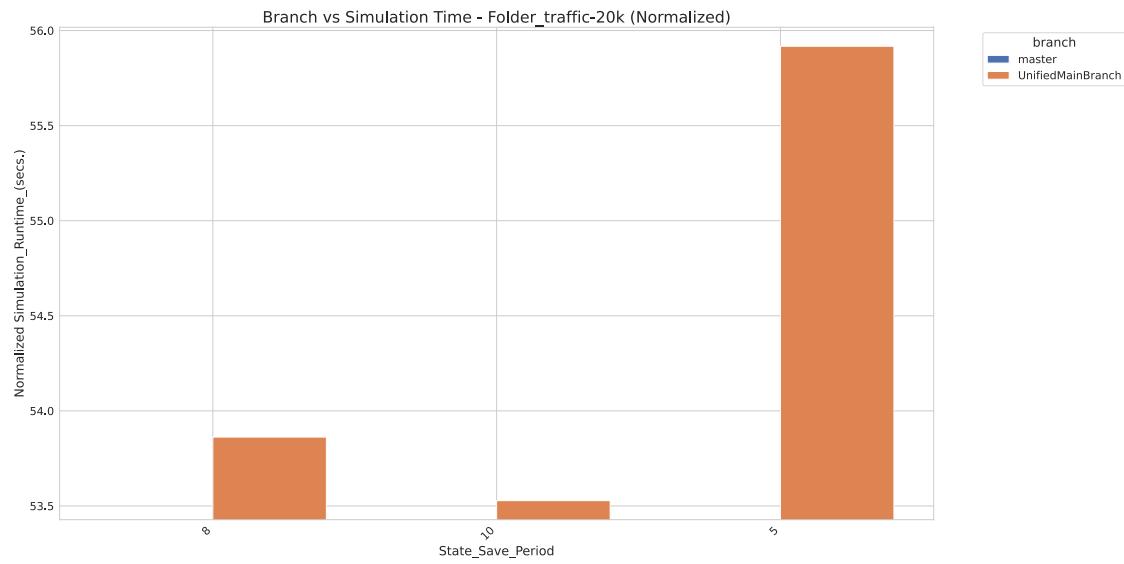


Figure 71: Branch vs Simulation Time_Folder_traffic-20k_normalized

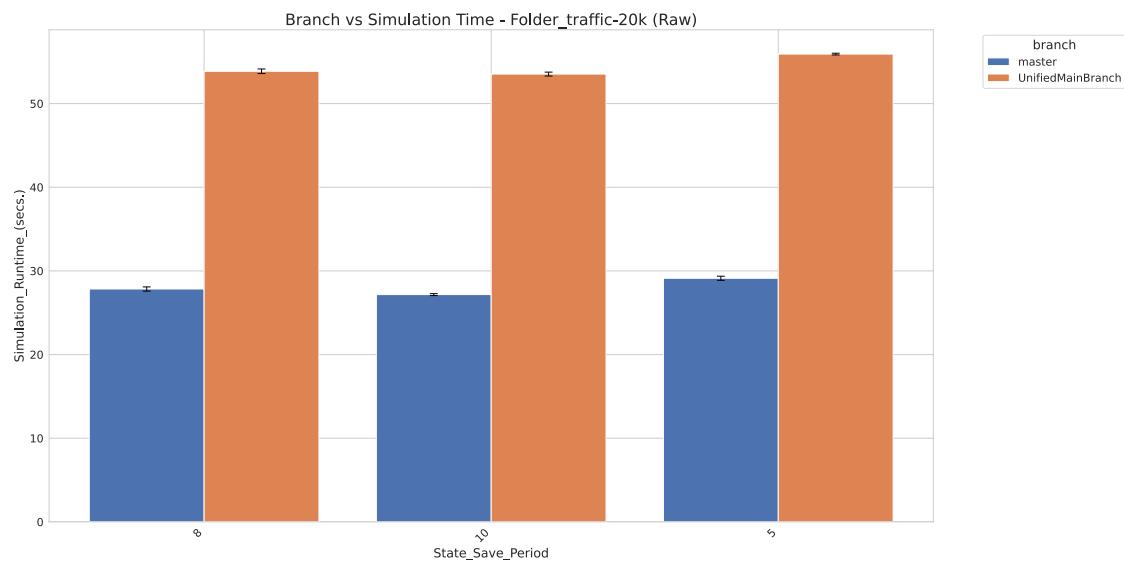


Figure 72: Branch vs Simulation Time_Folder_traffic-20k_raw

6 gaurav_unified_local

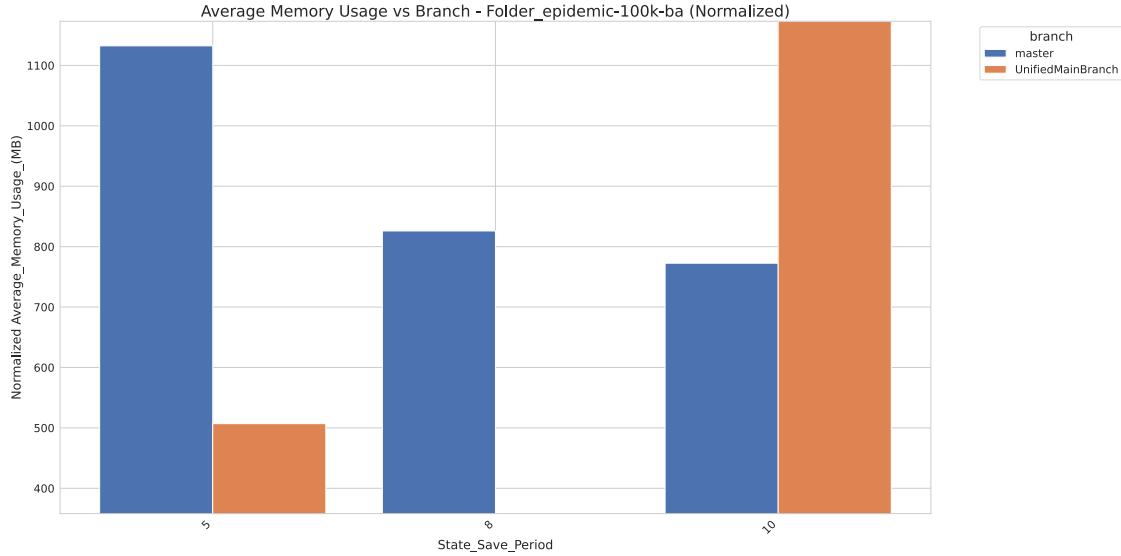


Figure 73: Average Memory Usage vs Branch.Folder.epidemic-100k-ba.normalized

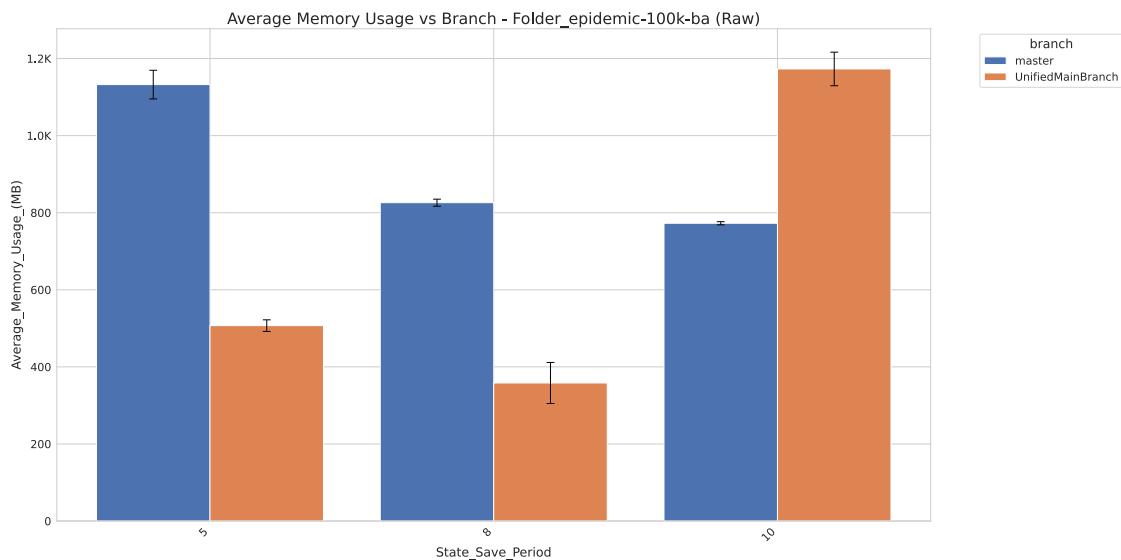


Figure 74: Average Memory Usage vs Branch.Folder.epidemic-100k-ba.raw

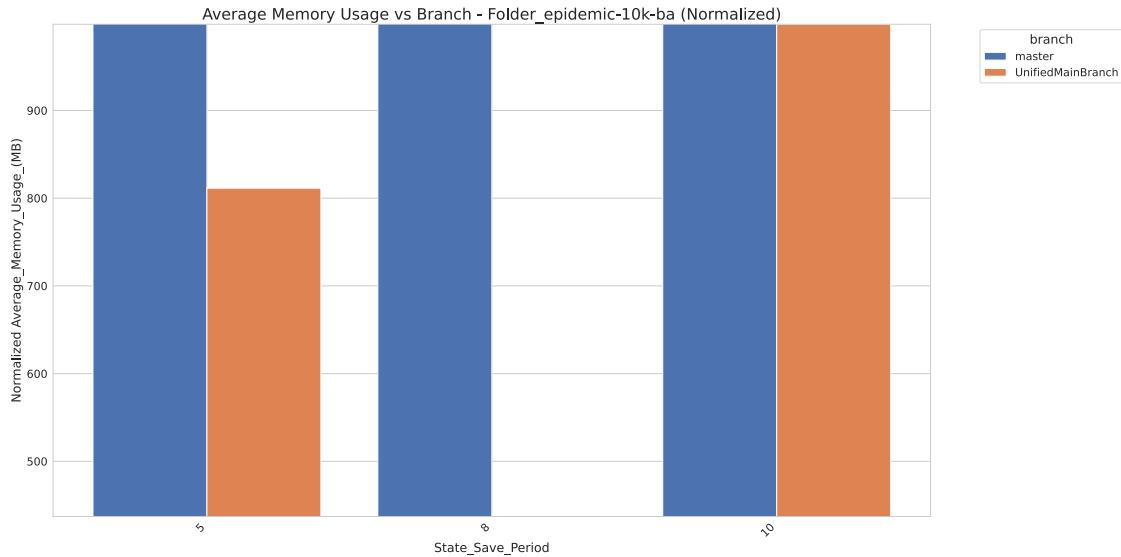


Figure 75: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

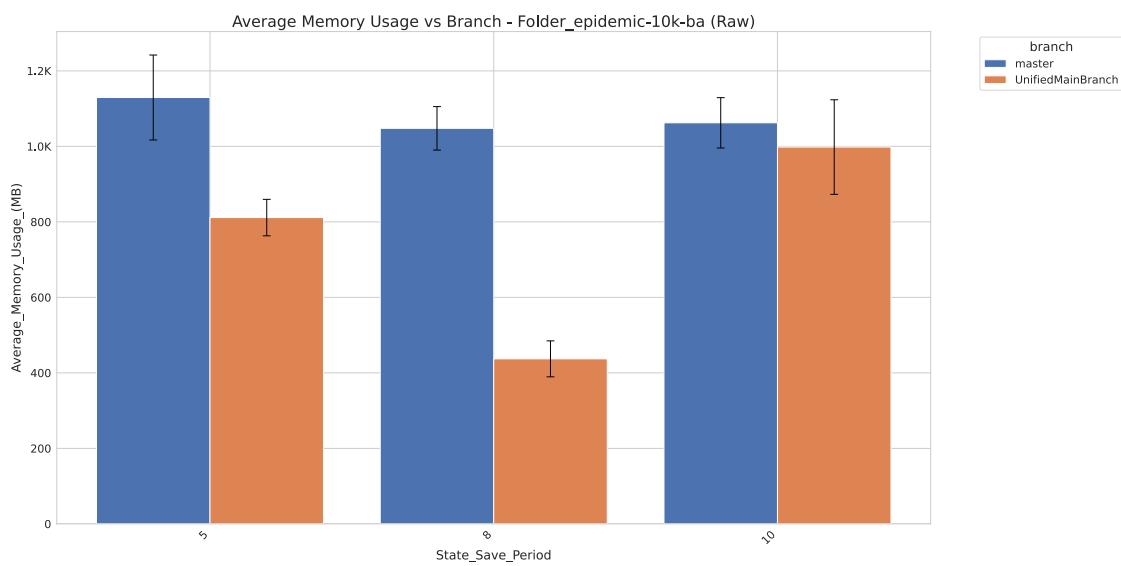


Figure 76: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

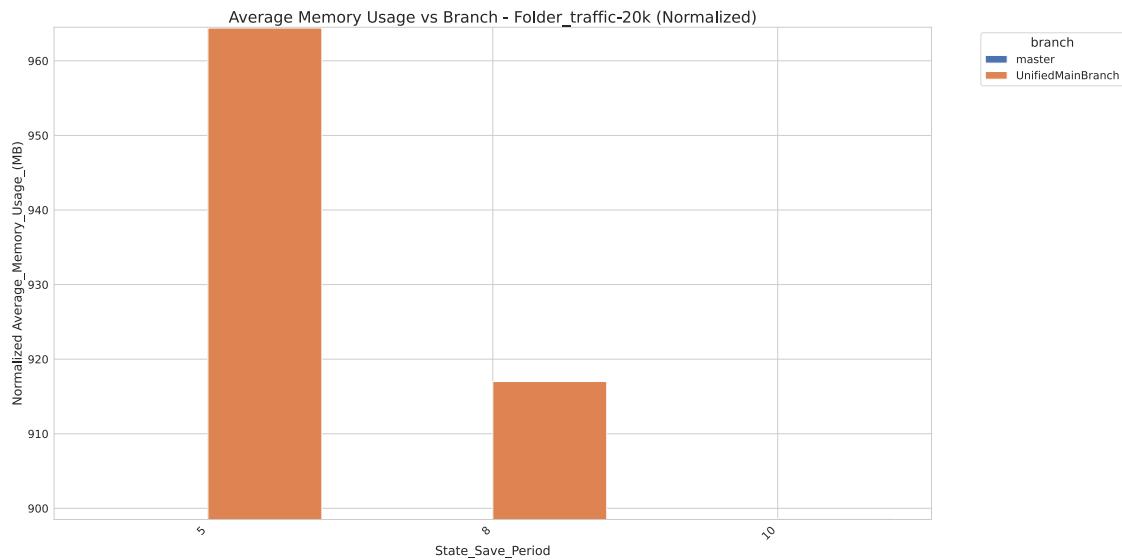


Figure 77: Average Memory Usage vs Branch_Folder_traffic-20k_normalized

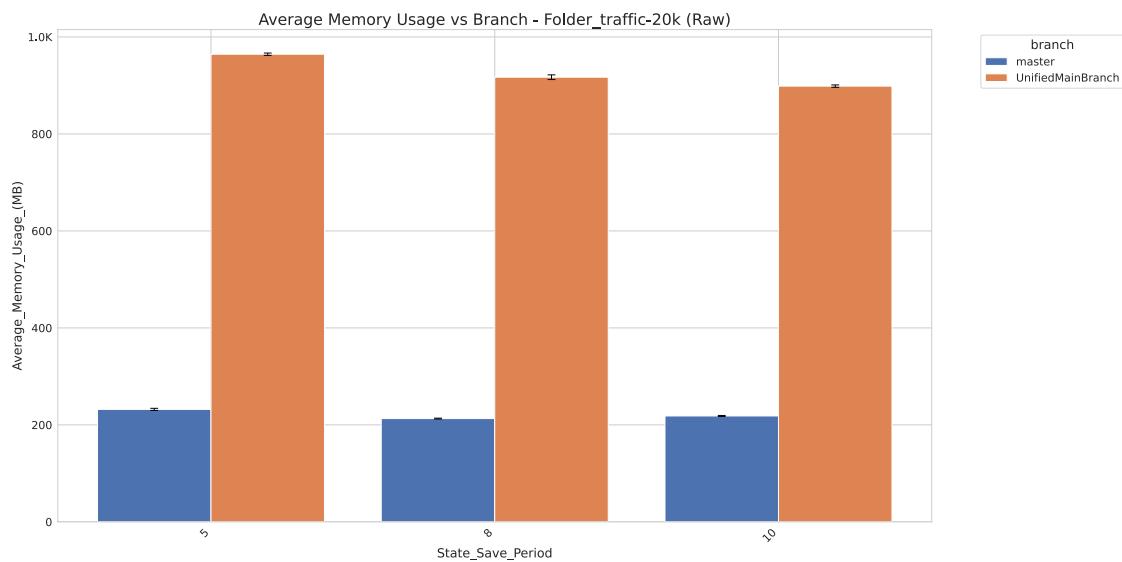


Figure 78: Average Memory Usage vs Branch_Folder_traffic-20k_raw

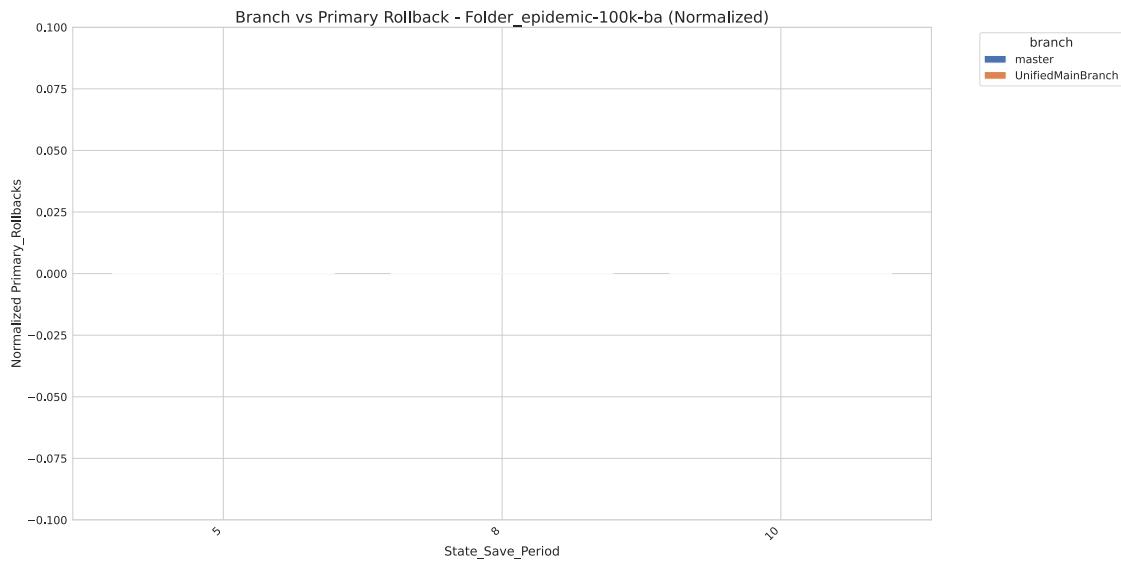


Figure 79: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

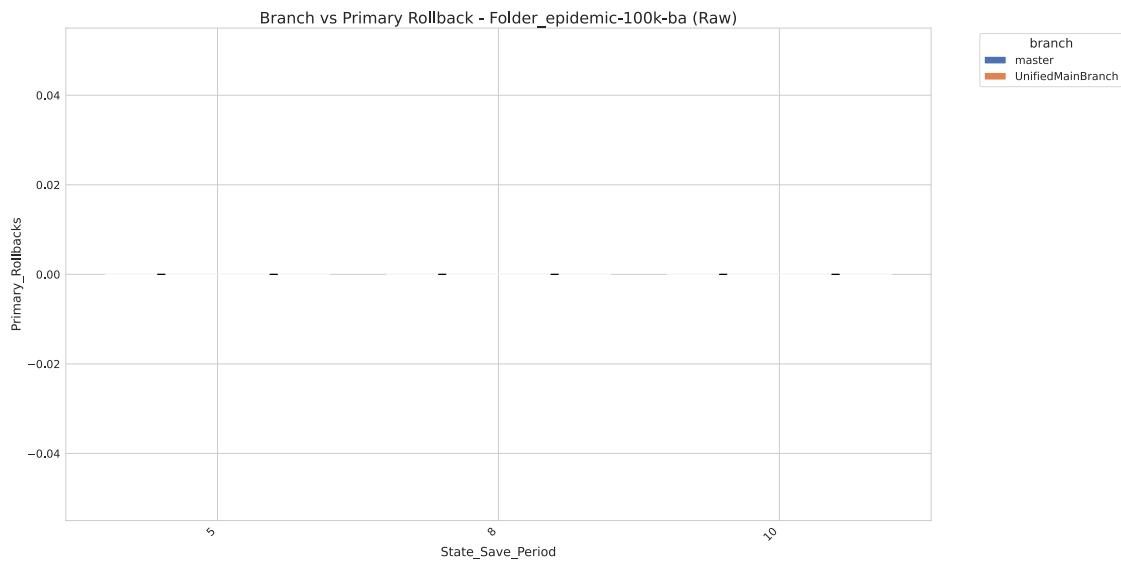


Figure 80: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

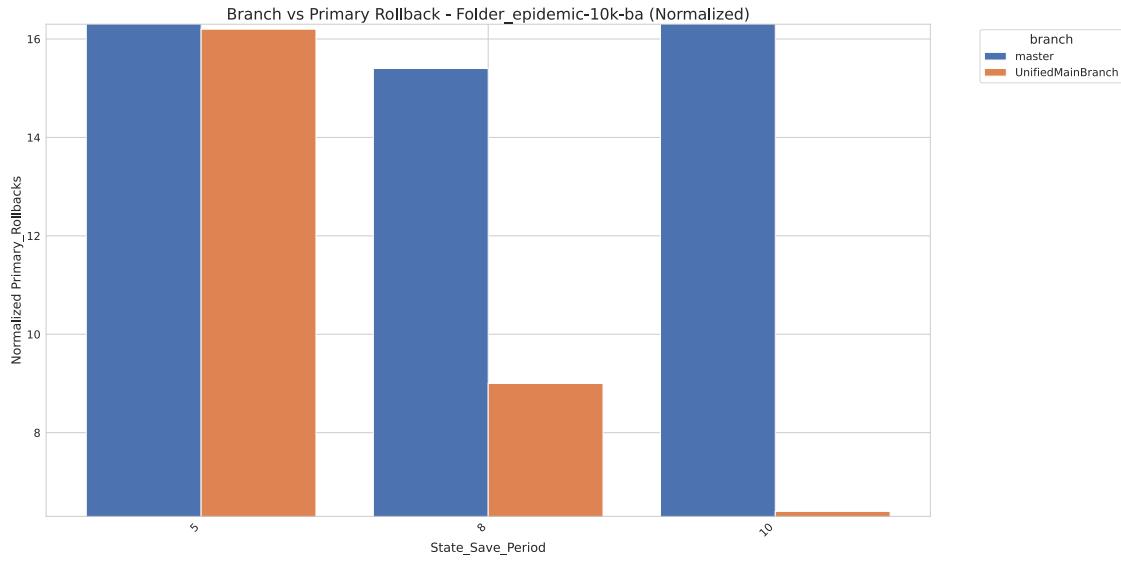


Figure 81: Branch vs Primary Rollback_Folder_epidemic-10k-ba_normalized

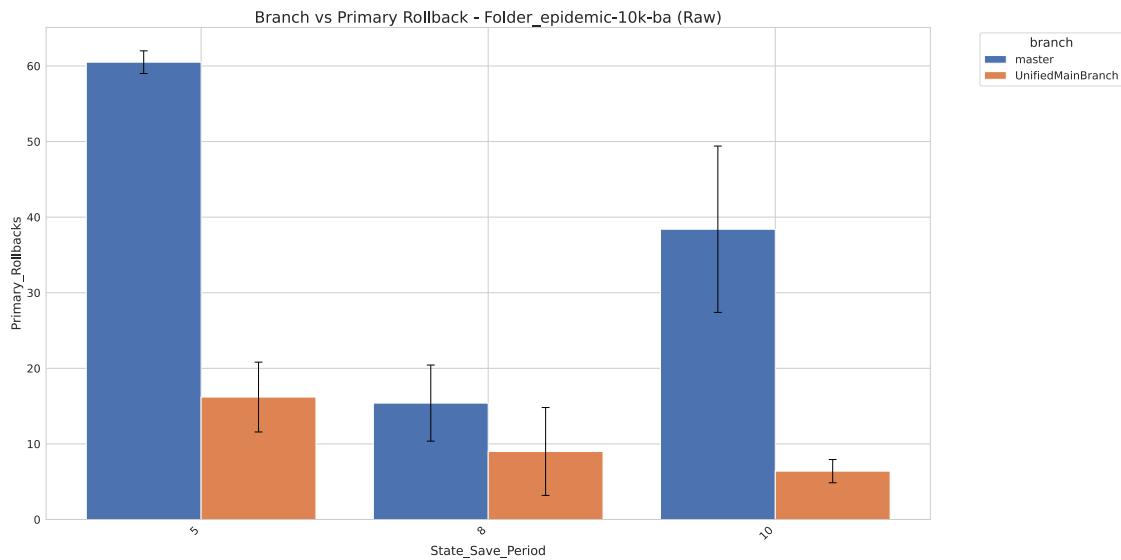


Figure 82: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

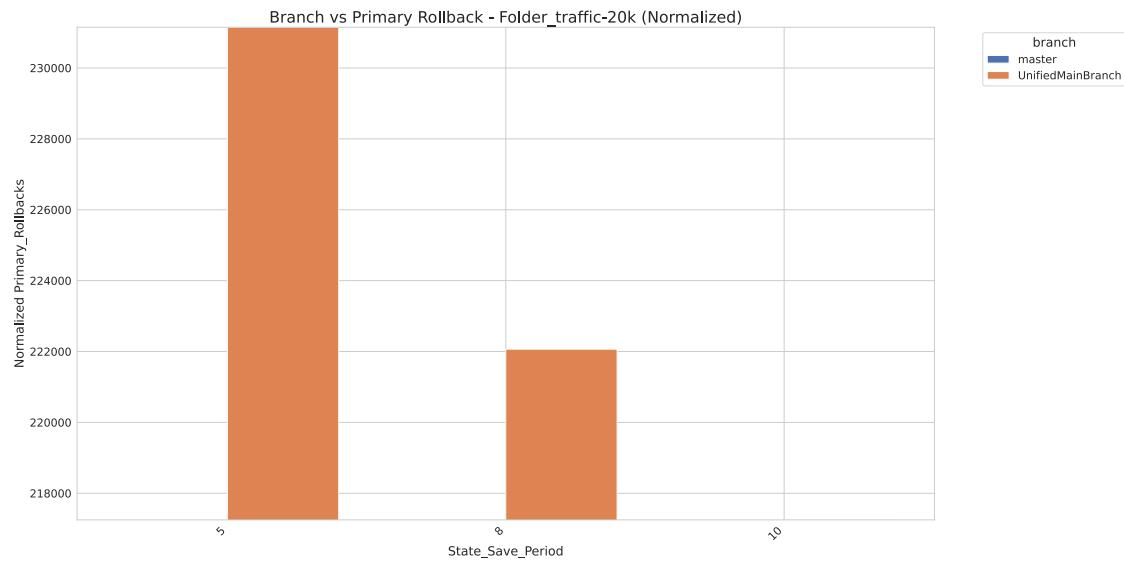


Figure 83: Branch vs Primary Rollback_Folder_traffic-20k_normalized

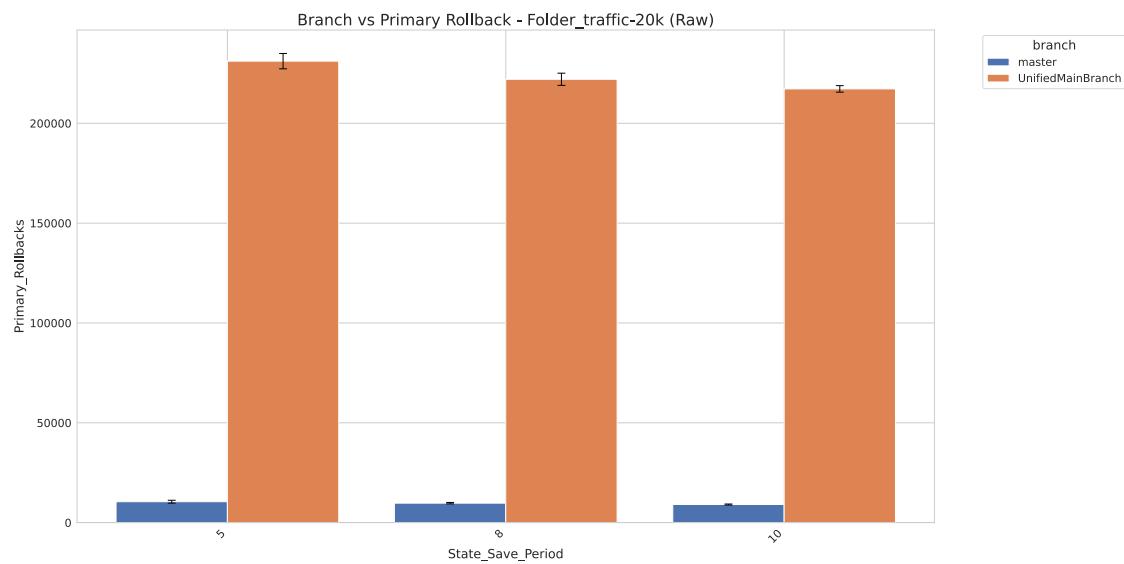


Figure 84: Branch vs Primary Rollback_Folder_traffic-20k_raw

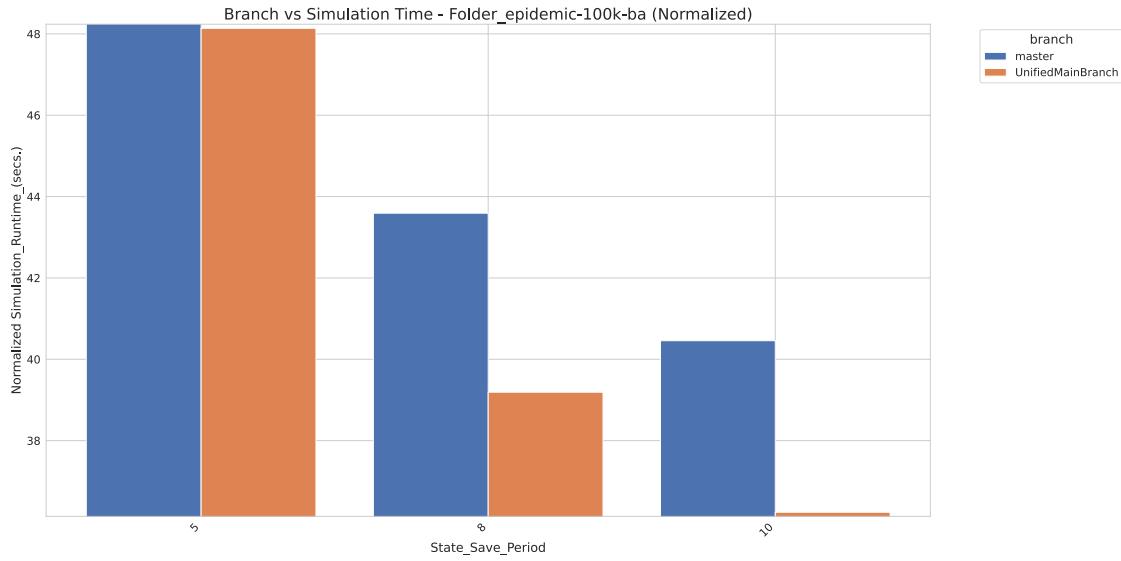


Figure 85: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

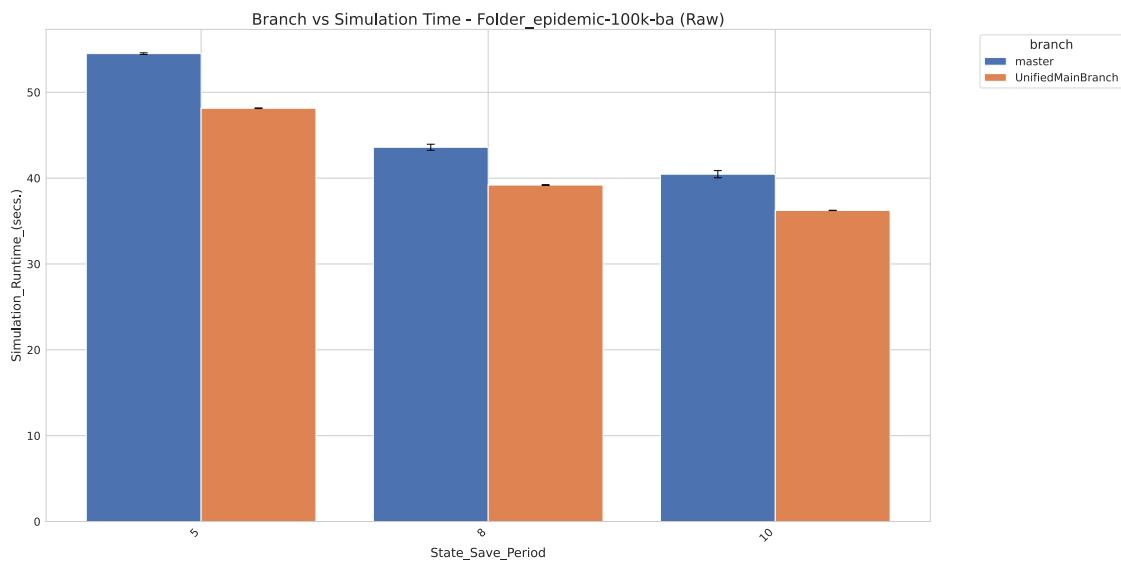


Figure 86: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

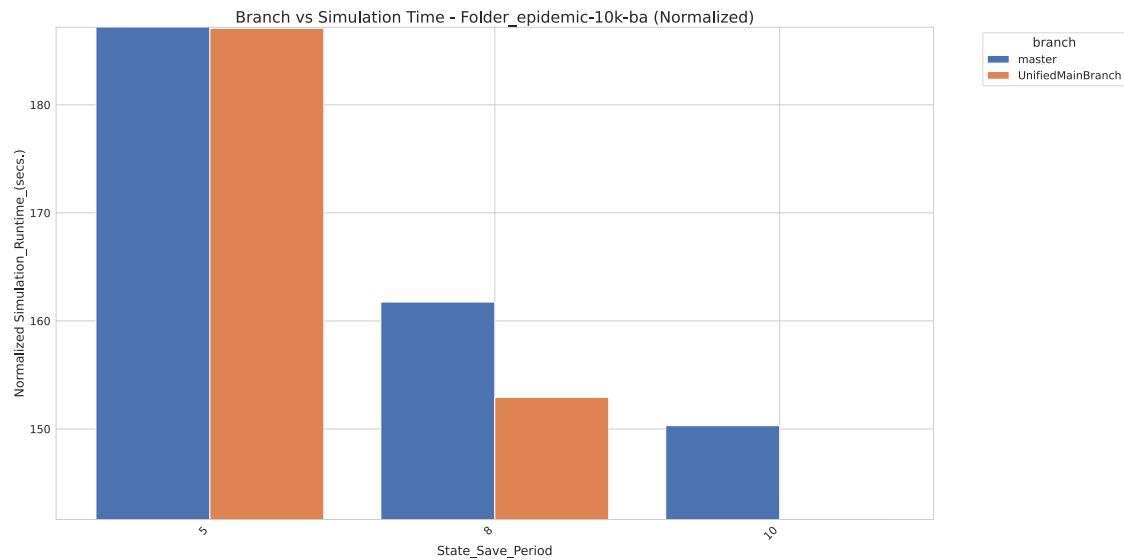


Figure 87: Branch vs Simulation Time_Folder_e pidemic-10k-ba_normalized

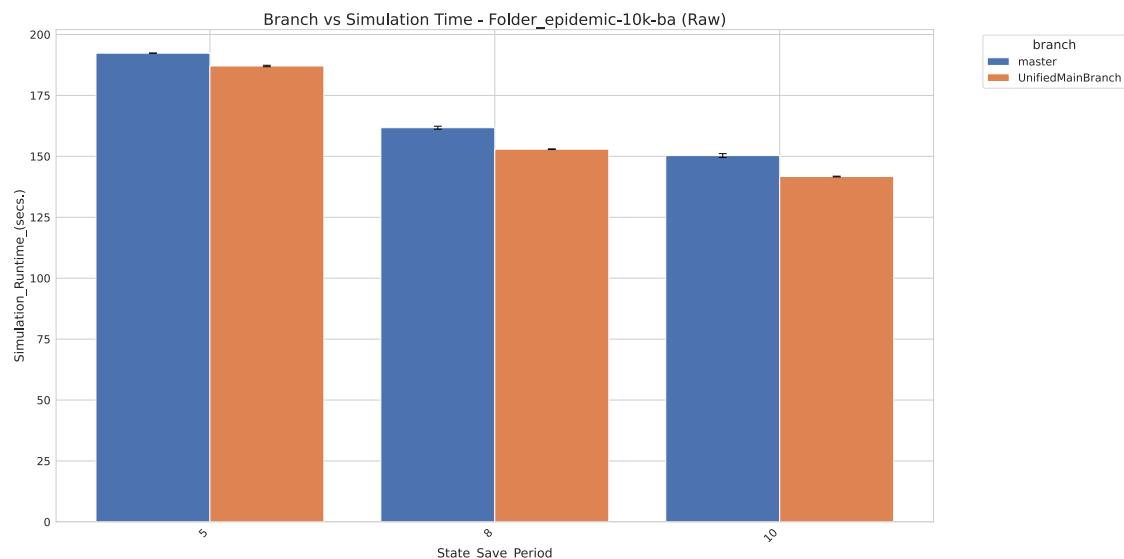


Figure 88: Branch vs Simulation Time_Folder_e pidemic-10k-ba_raw

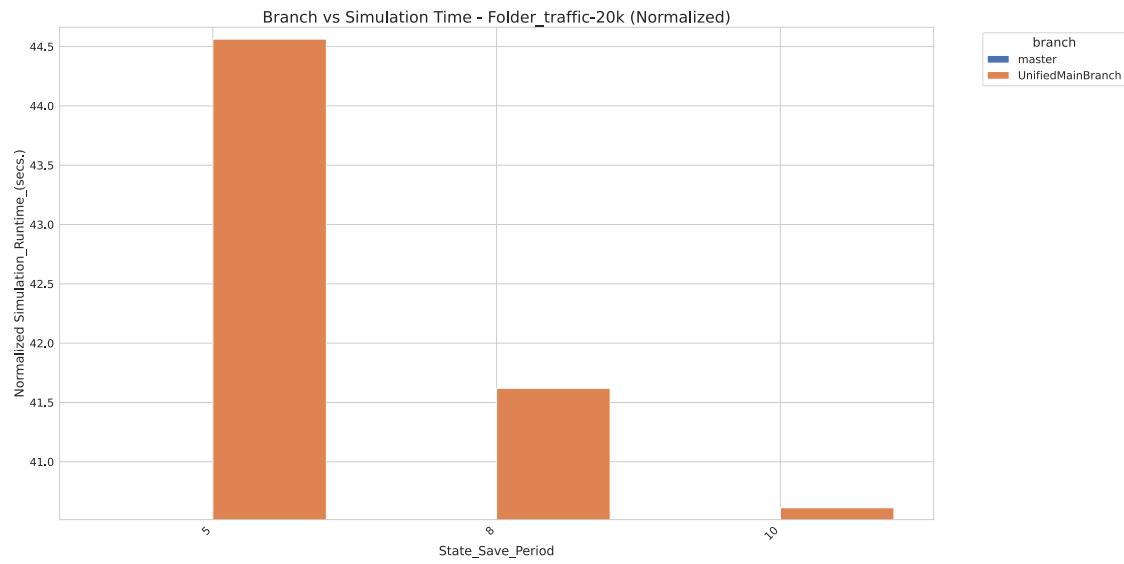


Figure 89: Branch vs Simulation Time_Folder_traffic-20k_normalized

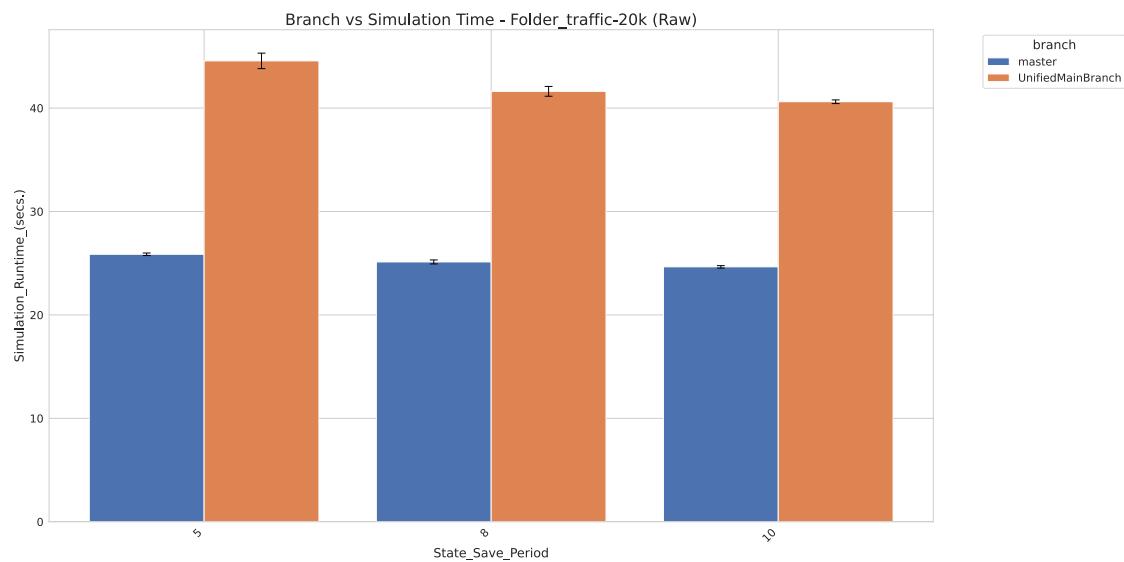


Figure 90: Branch vs Simulation Time_Folder_traffic-20k_raw

7 gvt_comparison

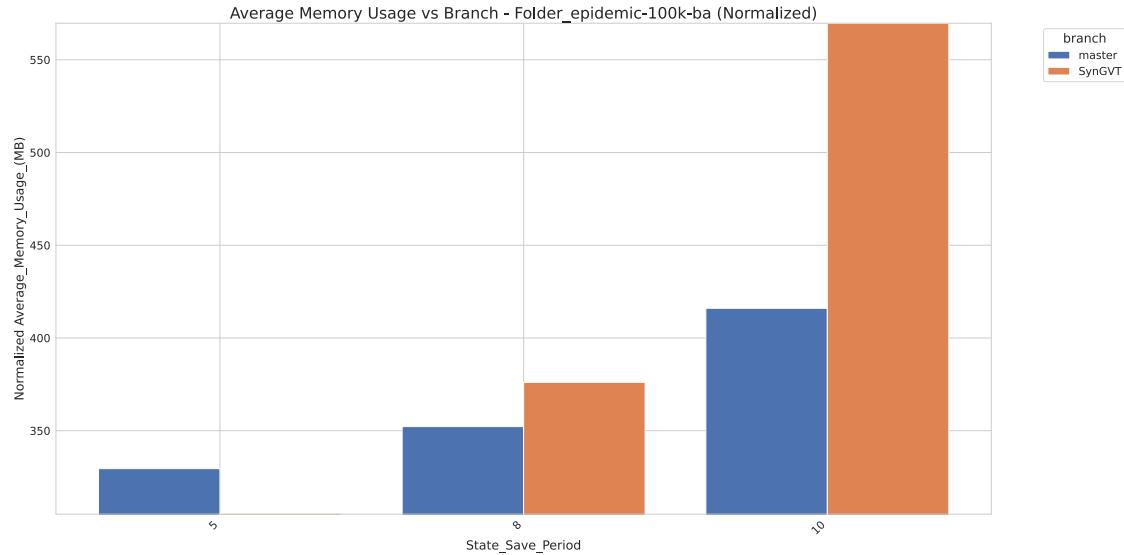


Figure 91: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

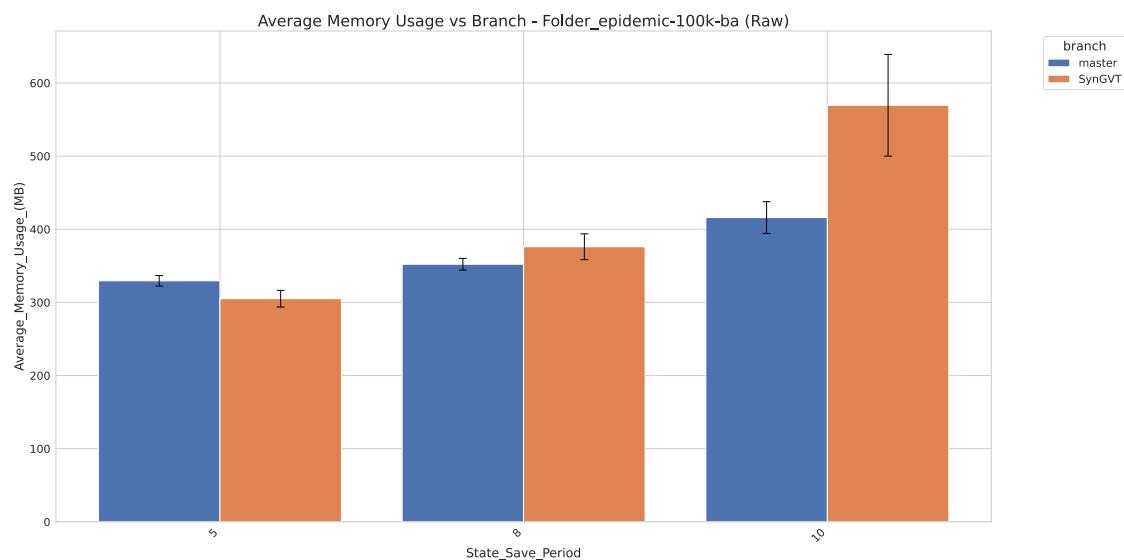


Figure 92: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

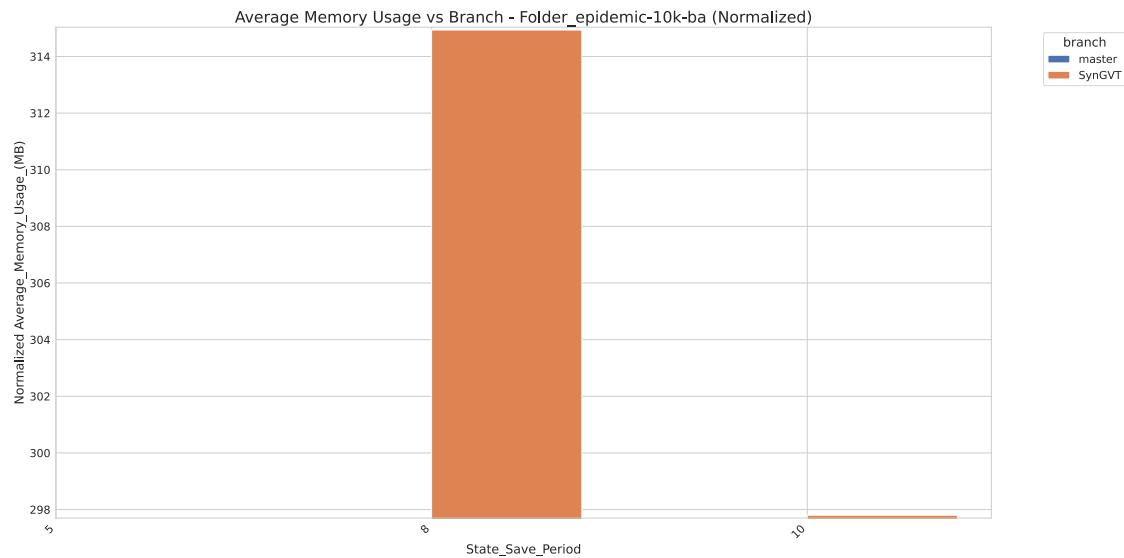


Figure 93: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

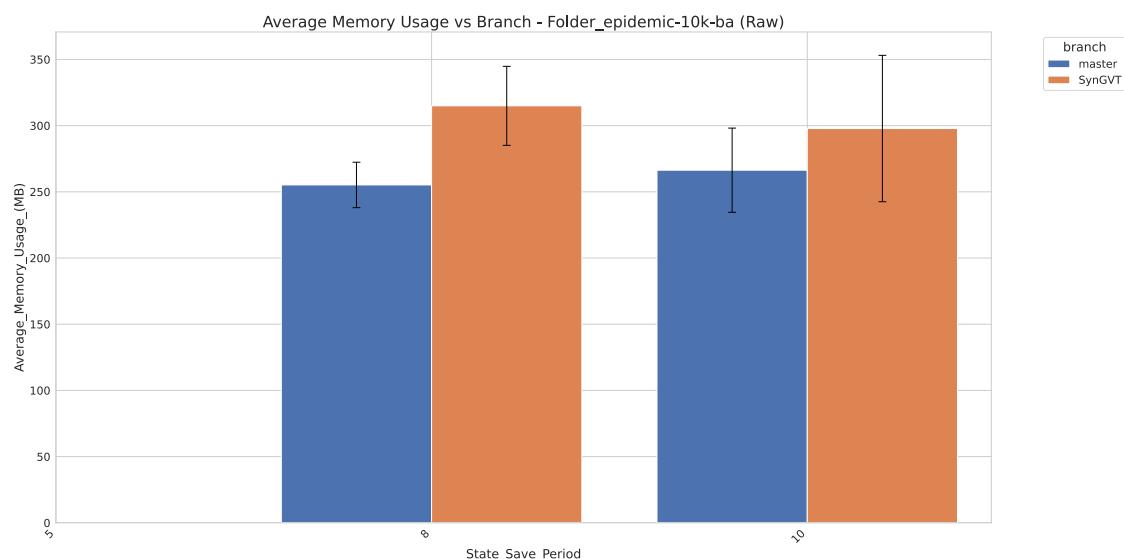


Figure 94: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

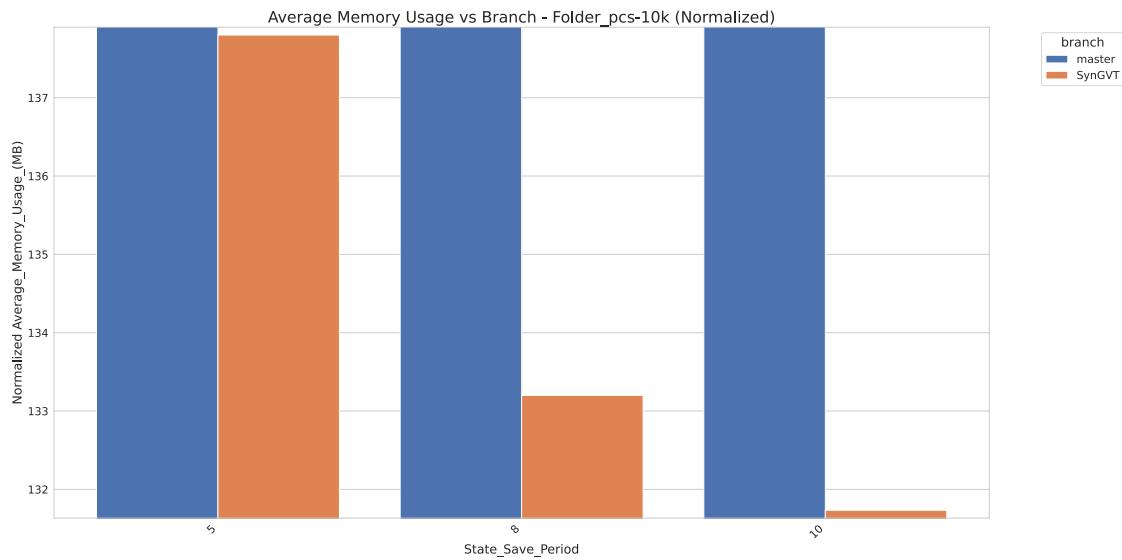


Figure 95: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

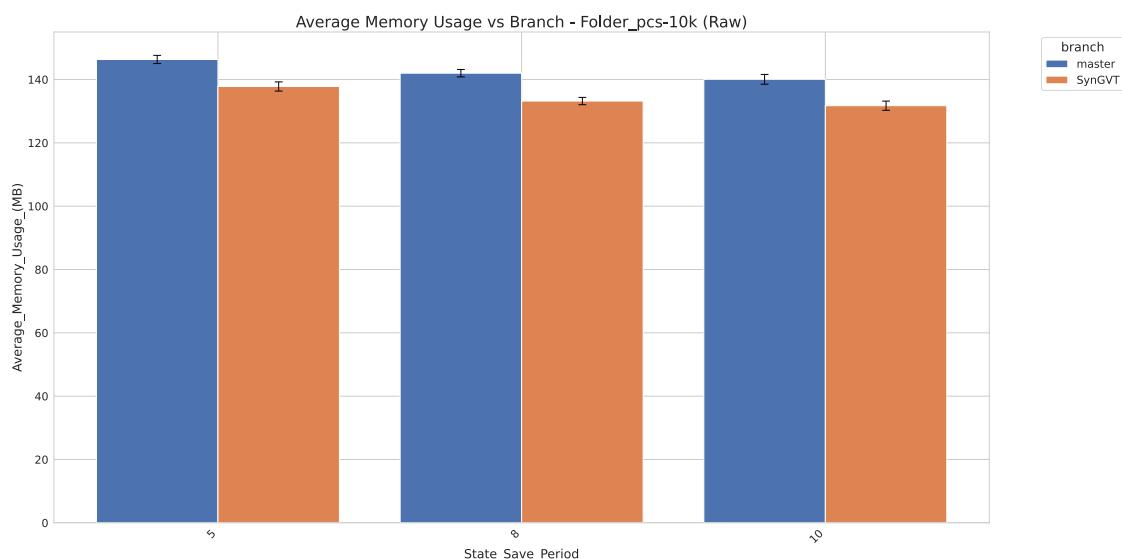


Figure 96: Average Memory Usage vs Branch_Folder_pcs-10k_raw

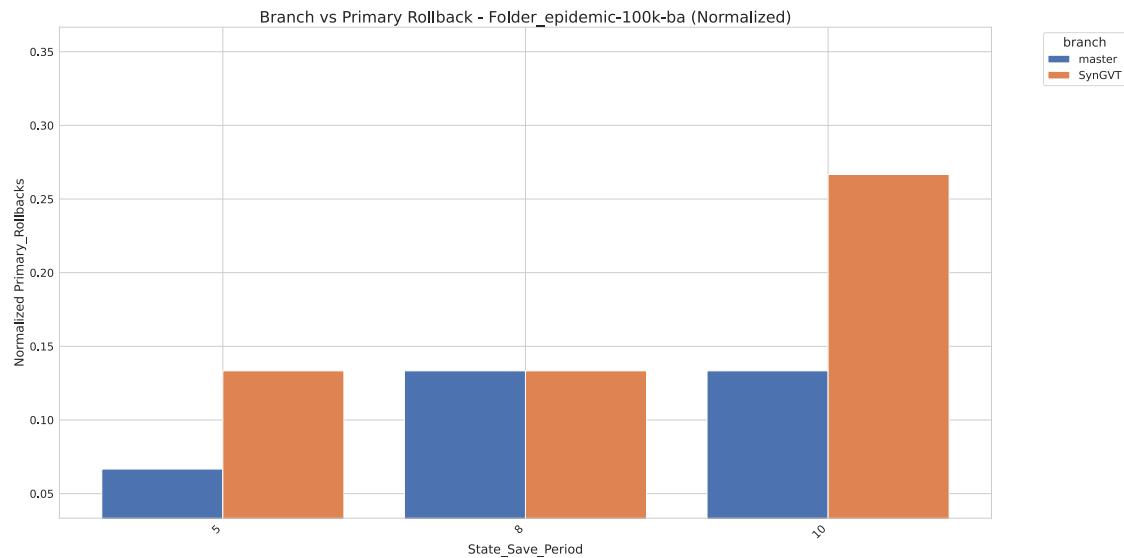


Figure 97: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

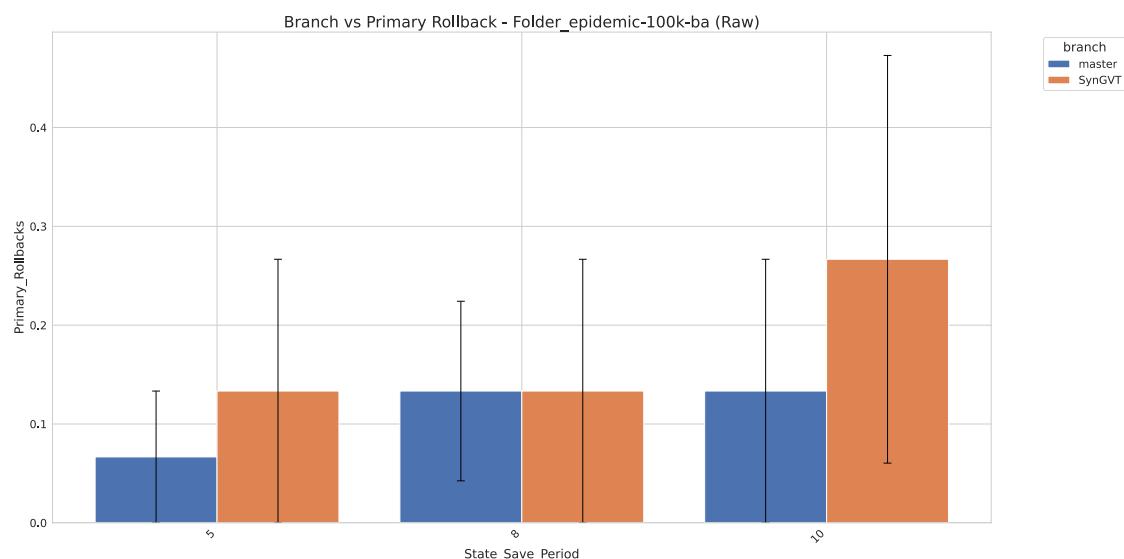


Figure 98: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

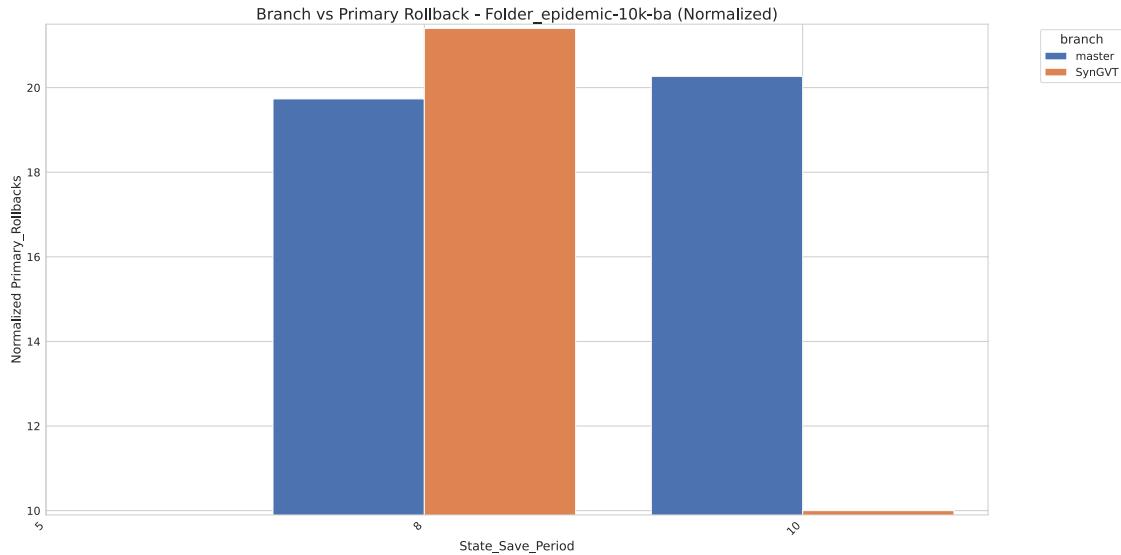


Figure 99: Branch vs Primary Rollback_Folder.epidemic-10k-ba_normalized

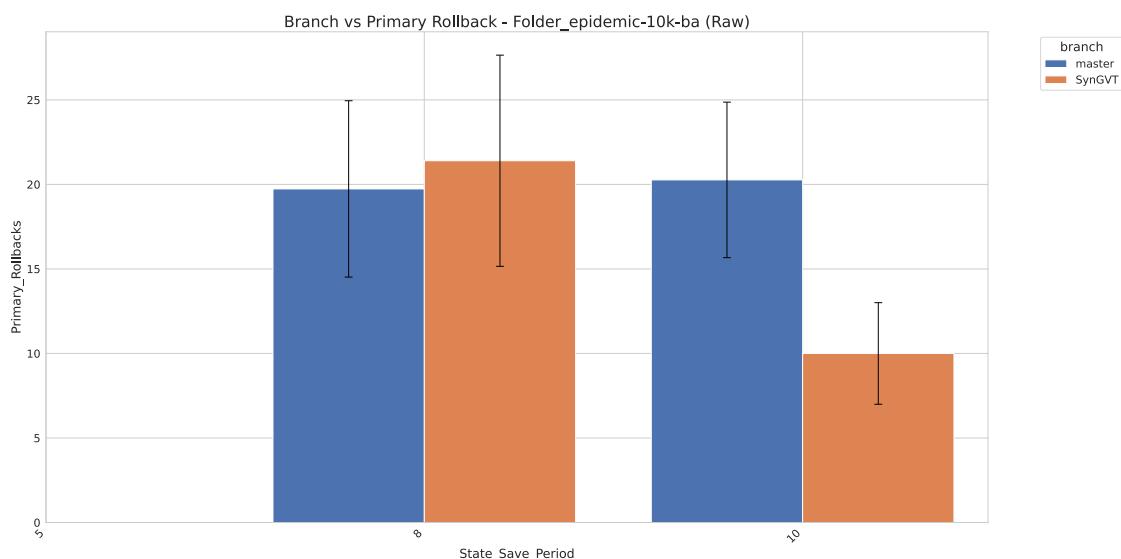


Figure 100: Branch vs Primary Rollback_Folder.epidemic-10k-ba_raw

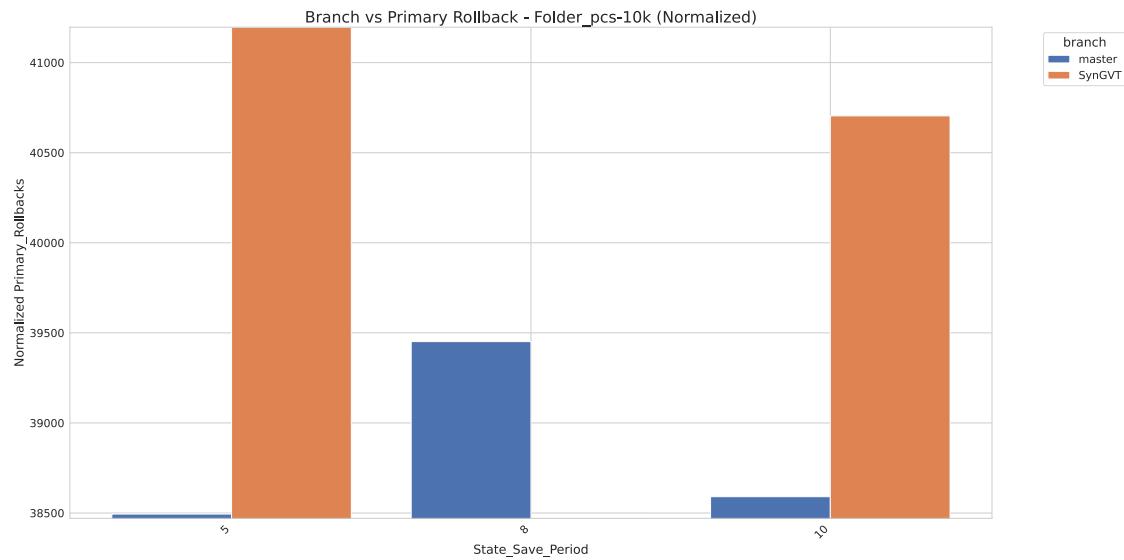


Figure 101: Branch vs Primary Rollback_Folder_pcs-10k_normalized

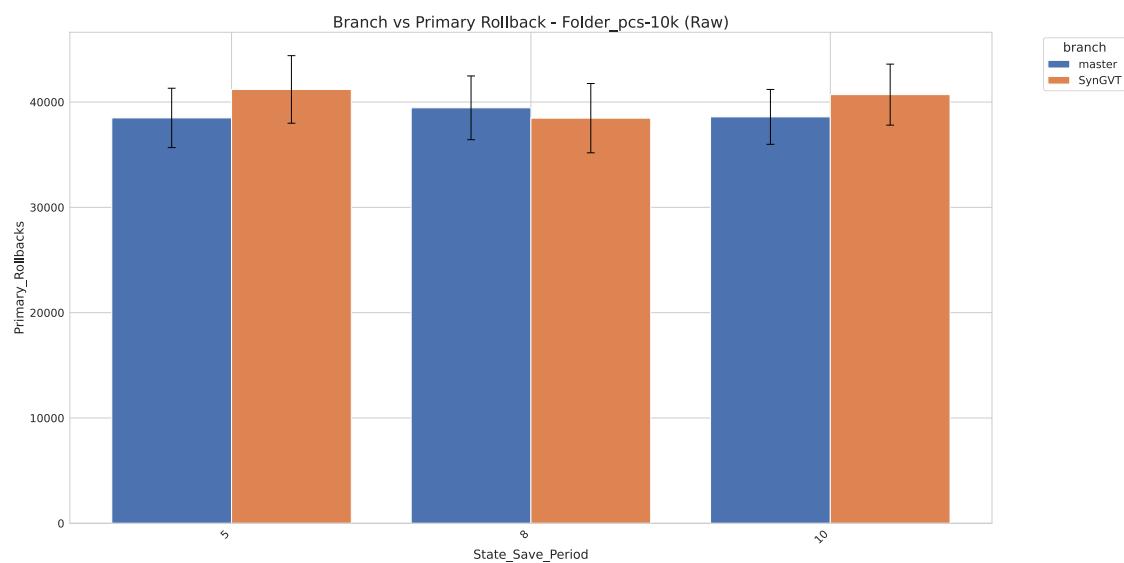


Figure 102: Branch vs Primary Rollback_Folder_pcs-10k_raw

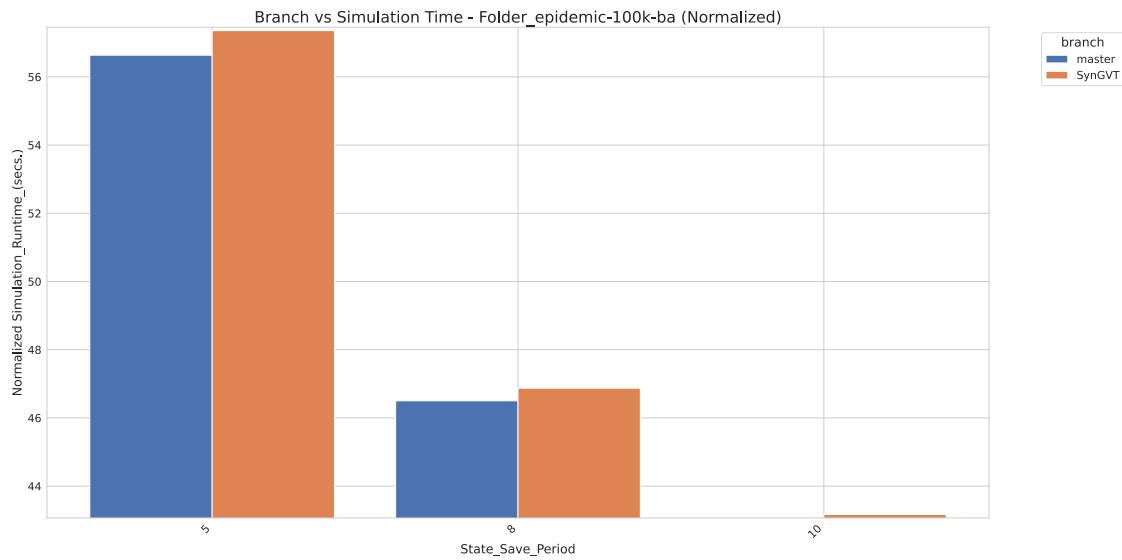


Figure 103: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

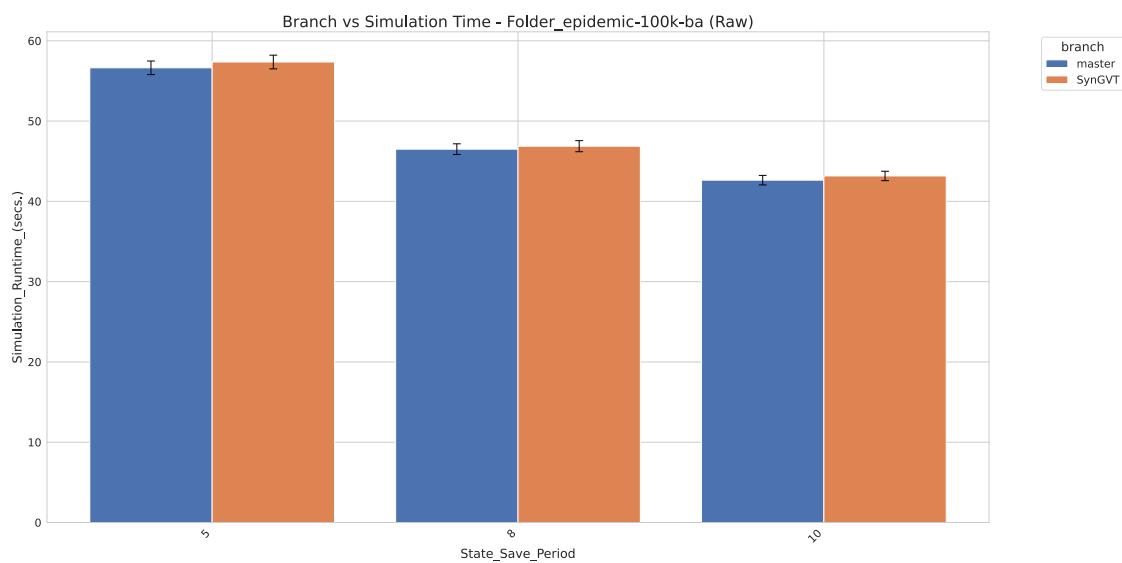


Figure 104: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

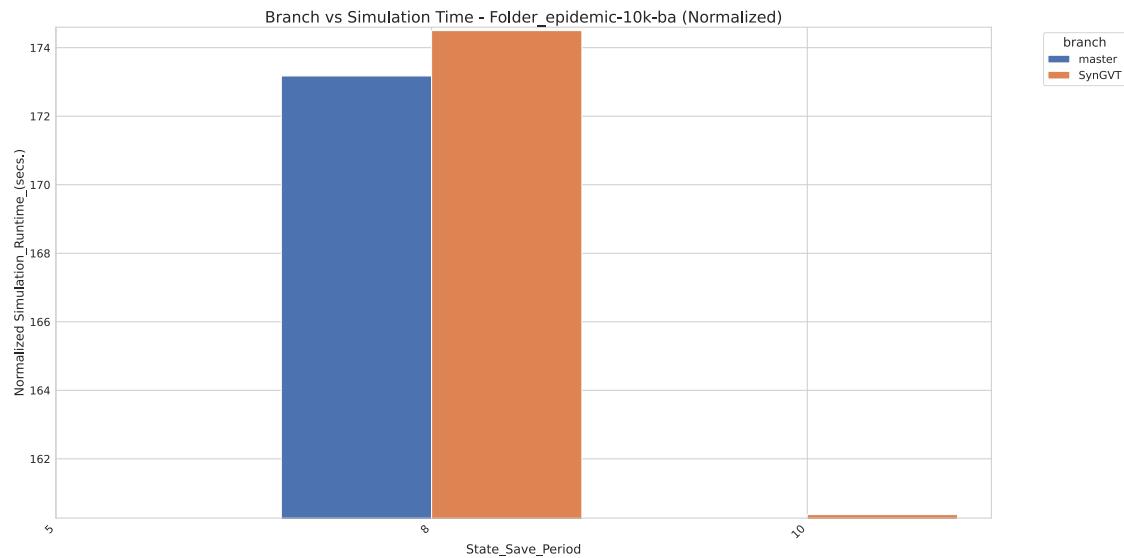


Figure 105: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

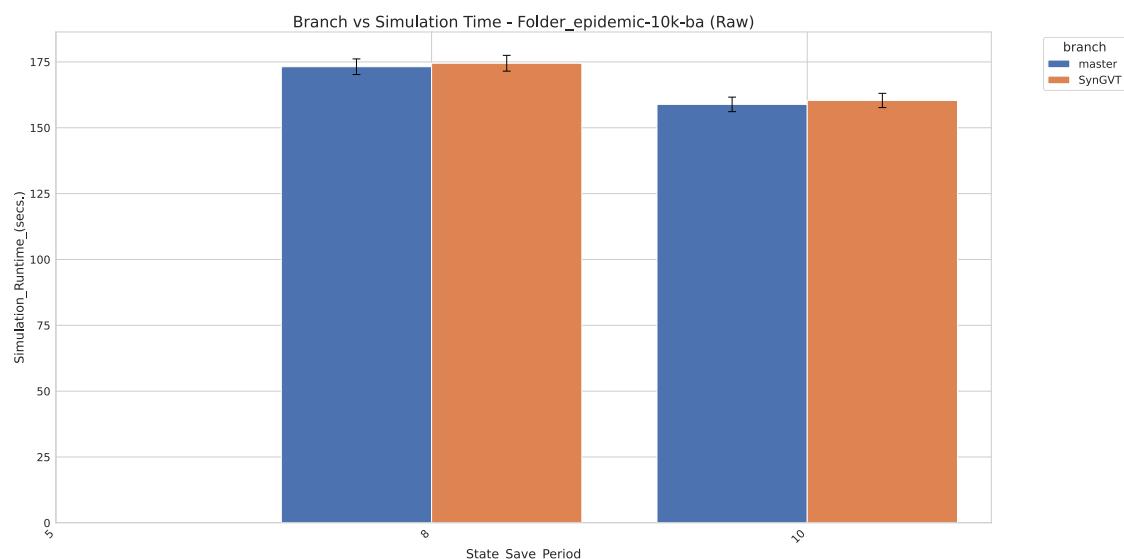


Figure 106: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

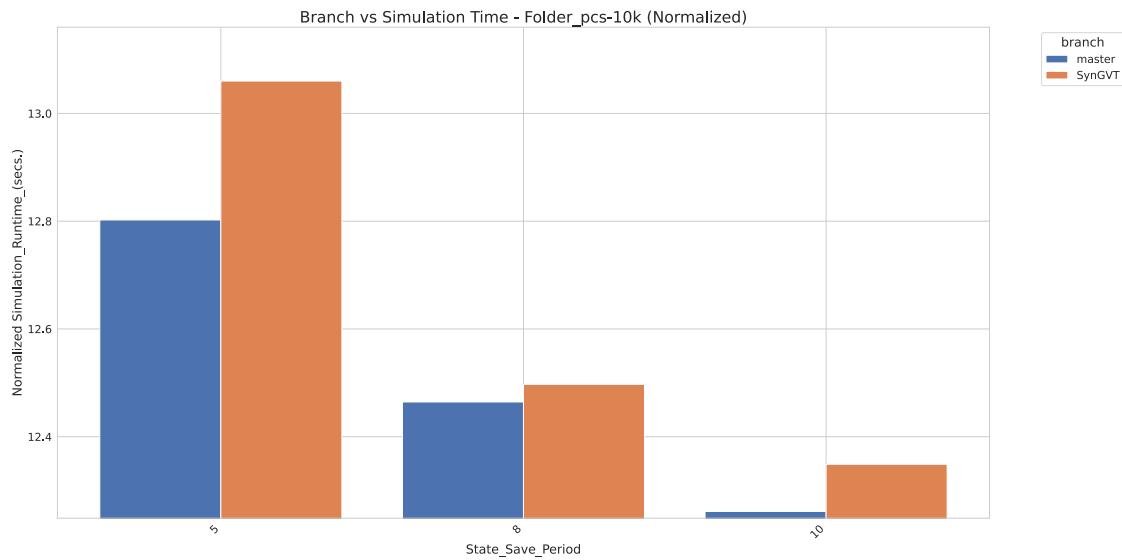


Figure 107: Branch vs Simulation Time_Folder_pcs-10k_normalized

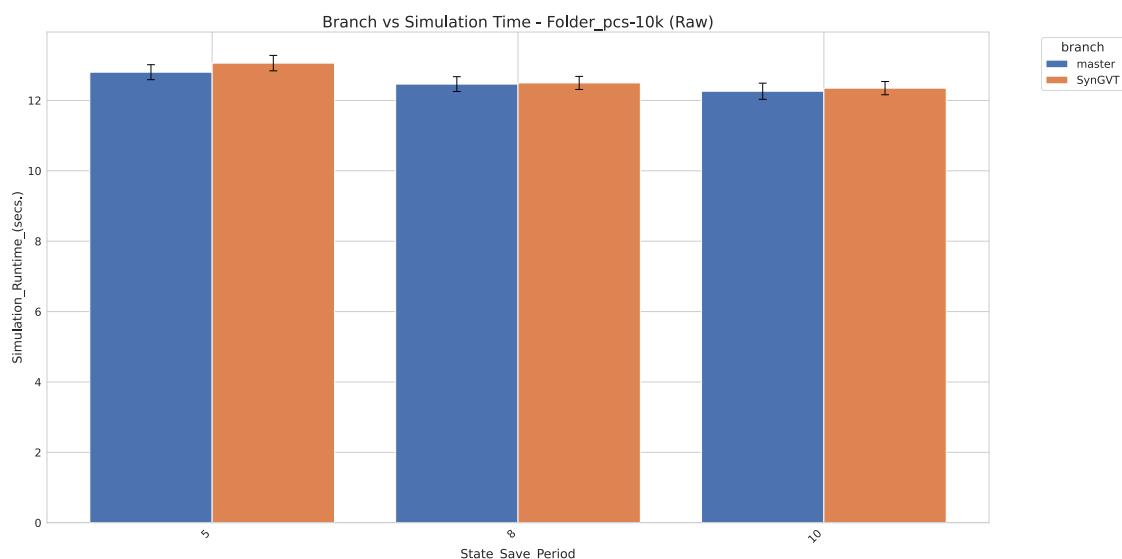


Figure 108: Branch vs Simulation Time_Folder_pcs-10k_raw

8 gvt_comparison_local

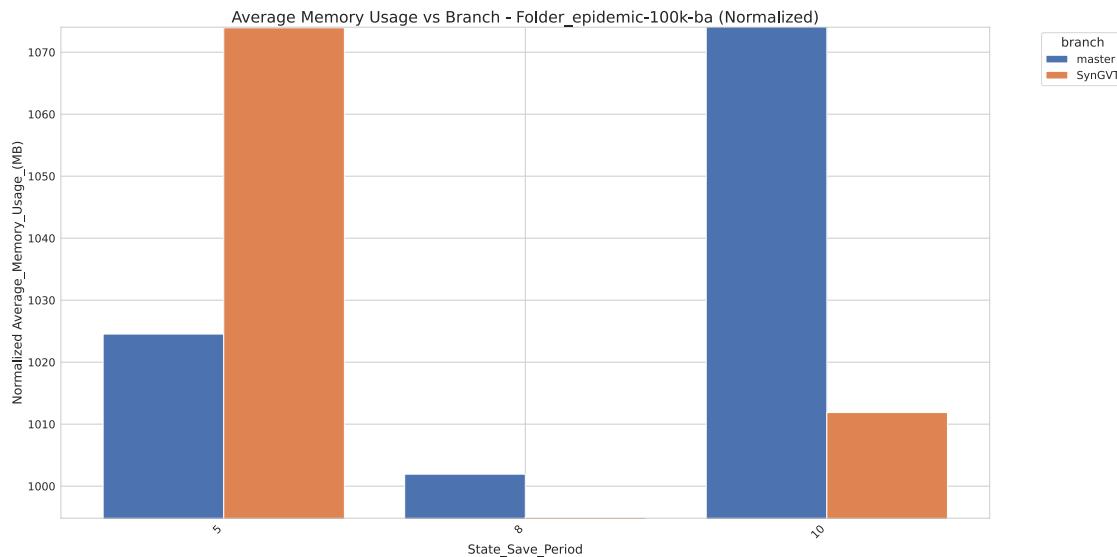


Figure 109: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

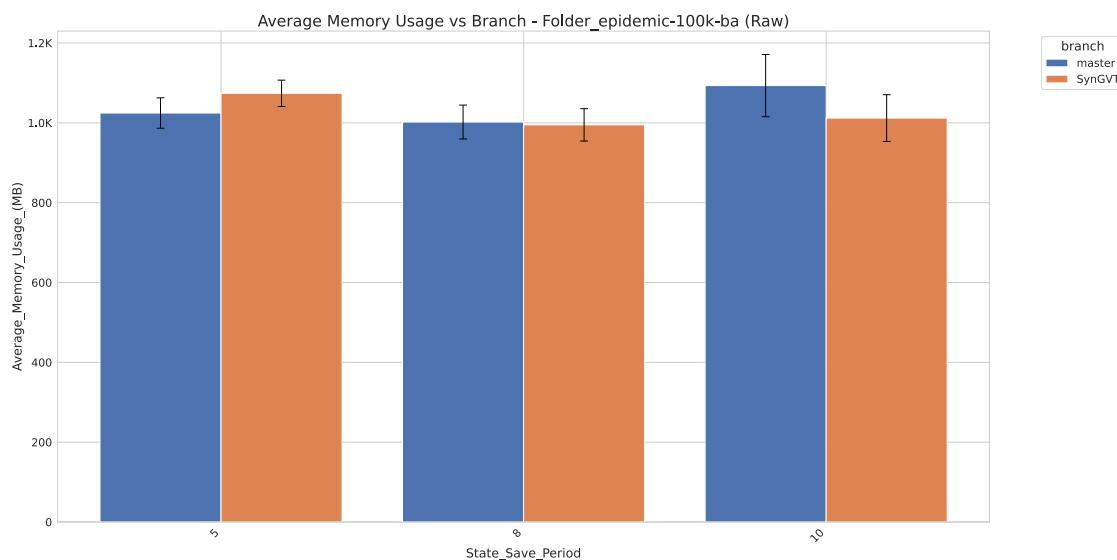


Figure 110: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

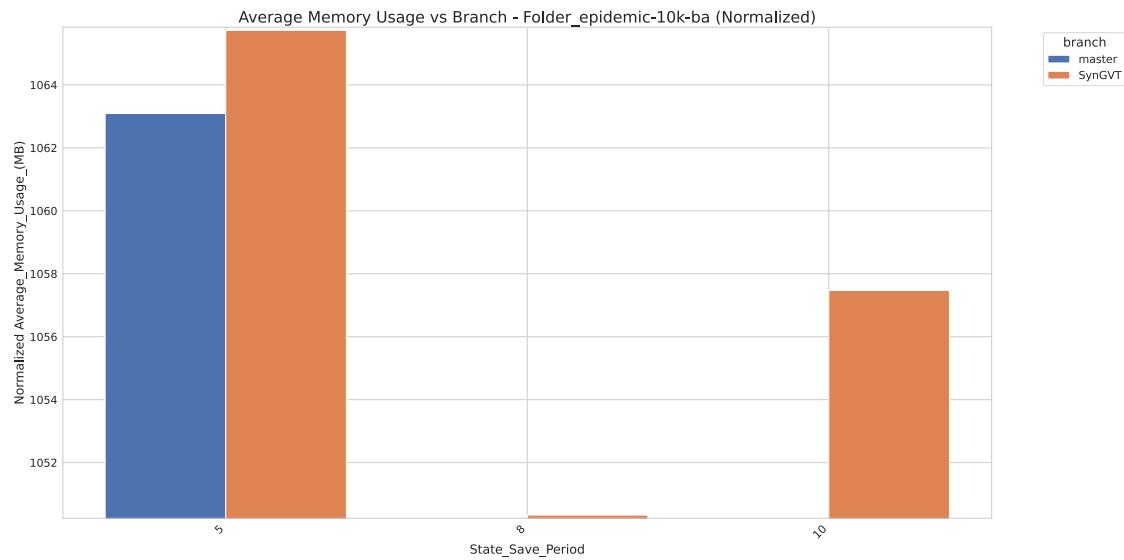


Figure 111: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

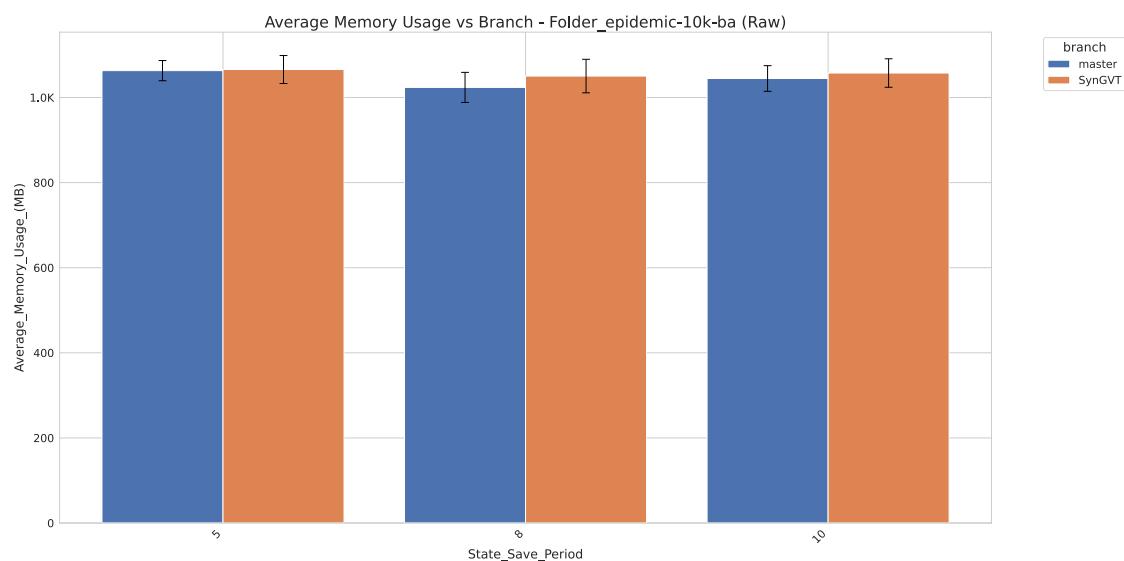


Figure 112: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

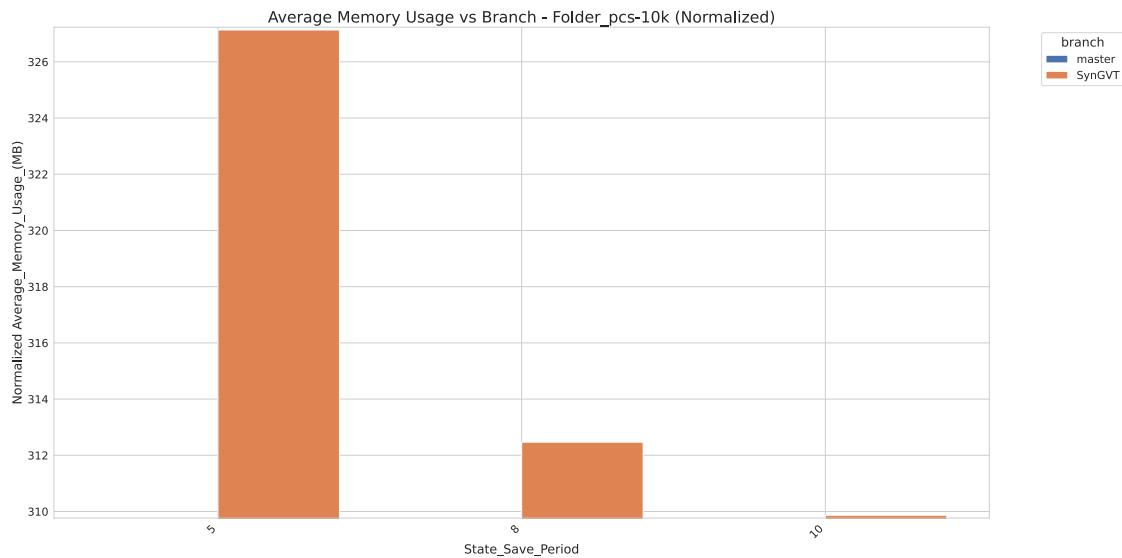


Figure 113: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

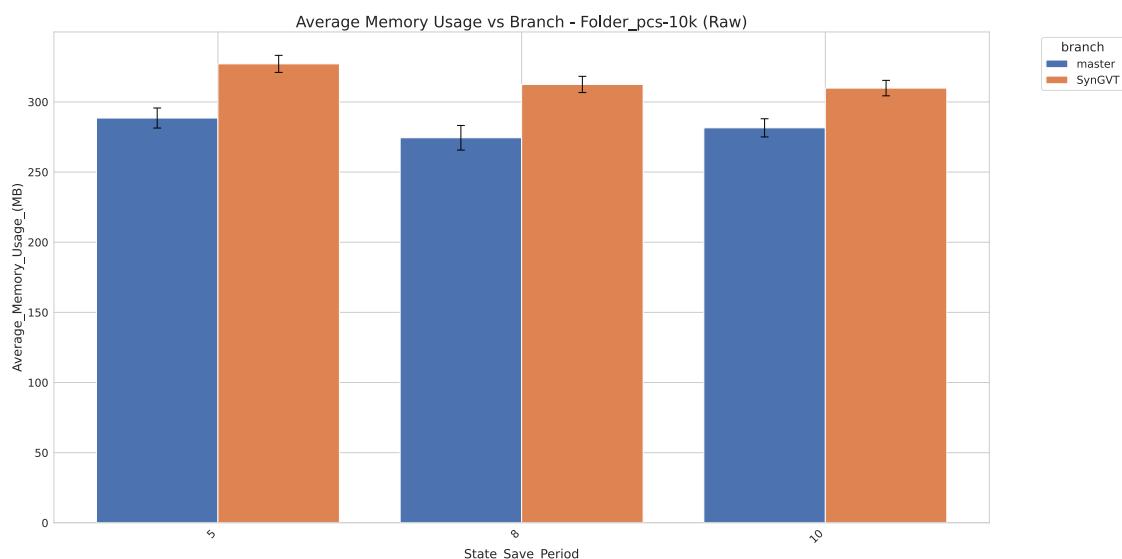


Figure 114: Average Memory Usage vs Branch_Folder_pcs-10k_raw

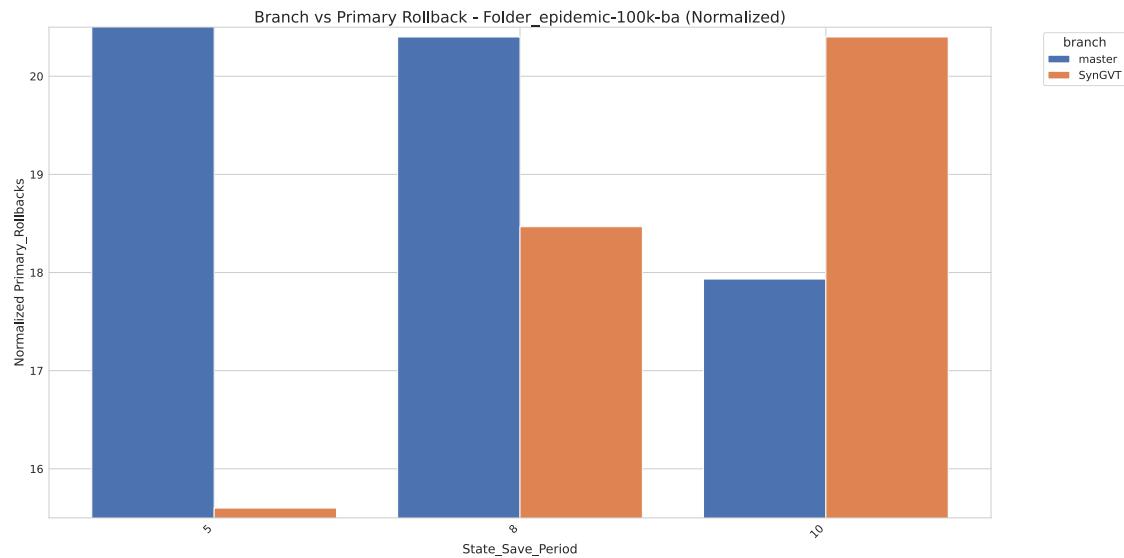


Figure 115: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

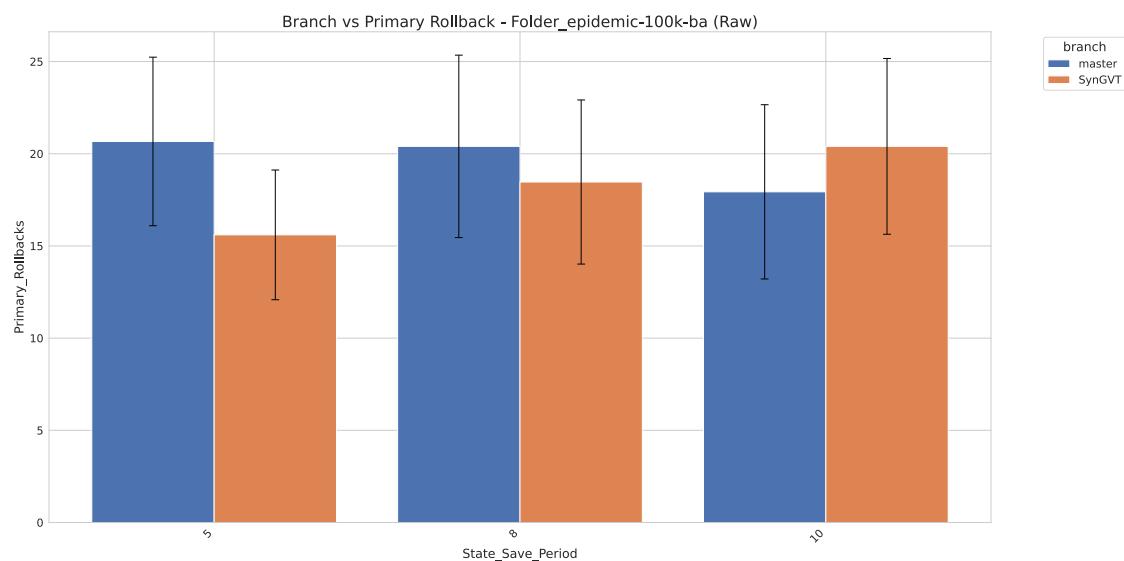


Figure 116: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

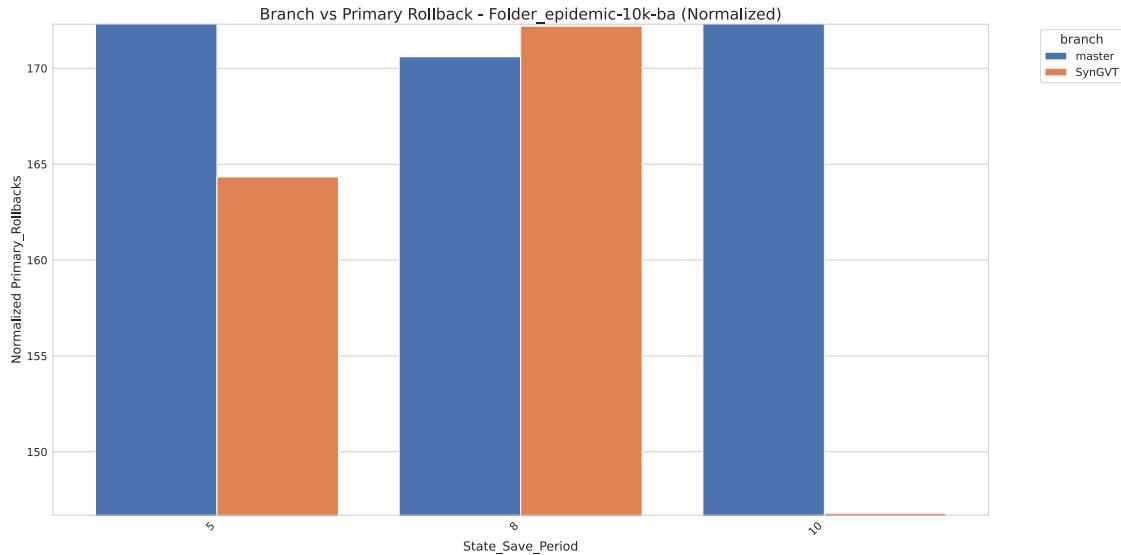


Figure 117: Branch vs Primary Rollback_Folder_epidemic-10k-ba_normalized

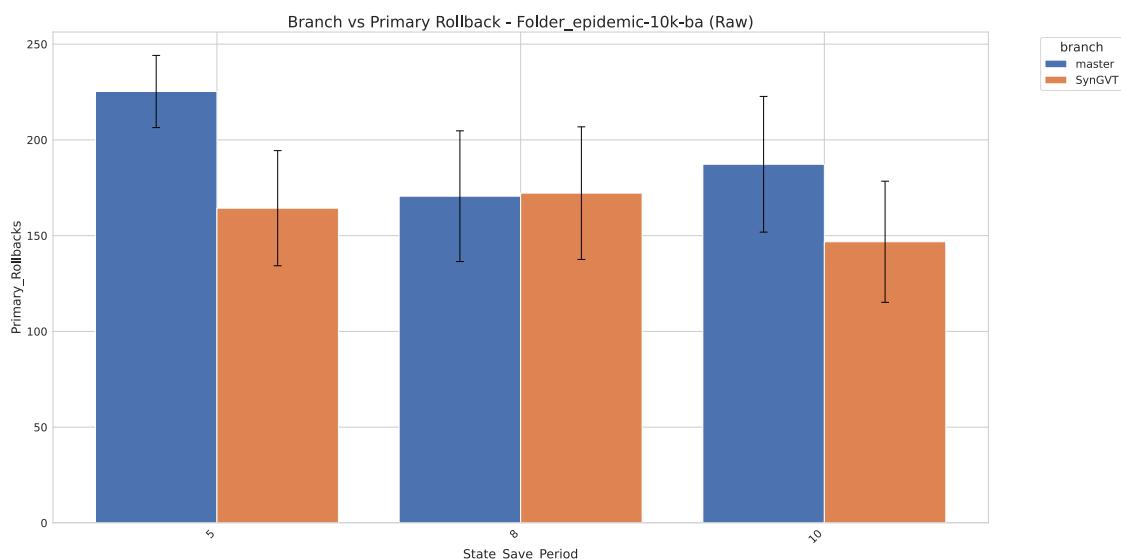


Figure 118: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

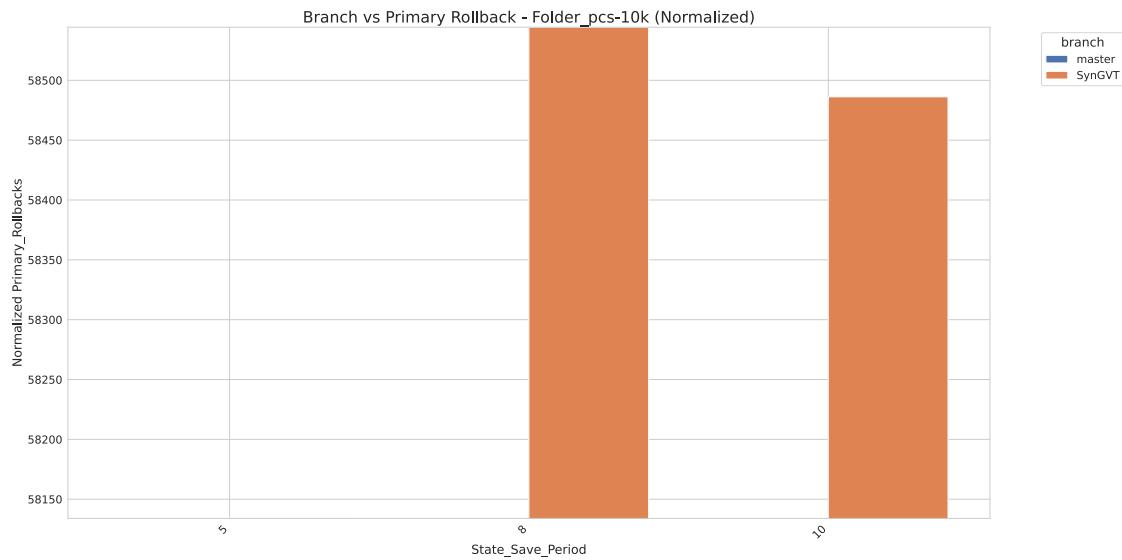


Figure 119: Branch vs Primary Rollback_Folder_pcs-10k_normalized

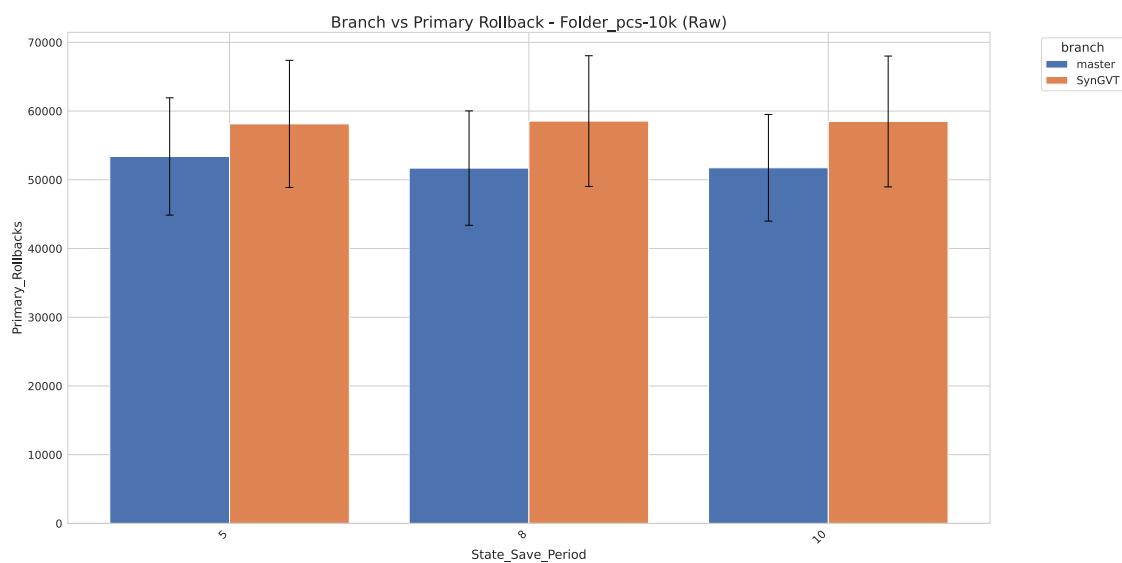


Figure 120: Branch vs Primary Rollback_Folder_pcs-10k_raw

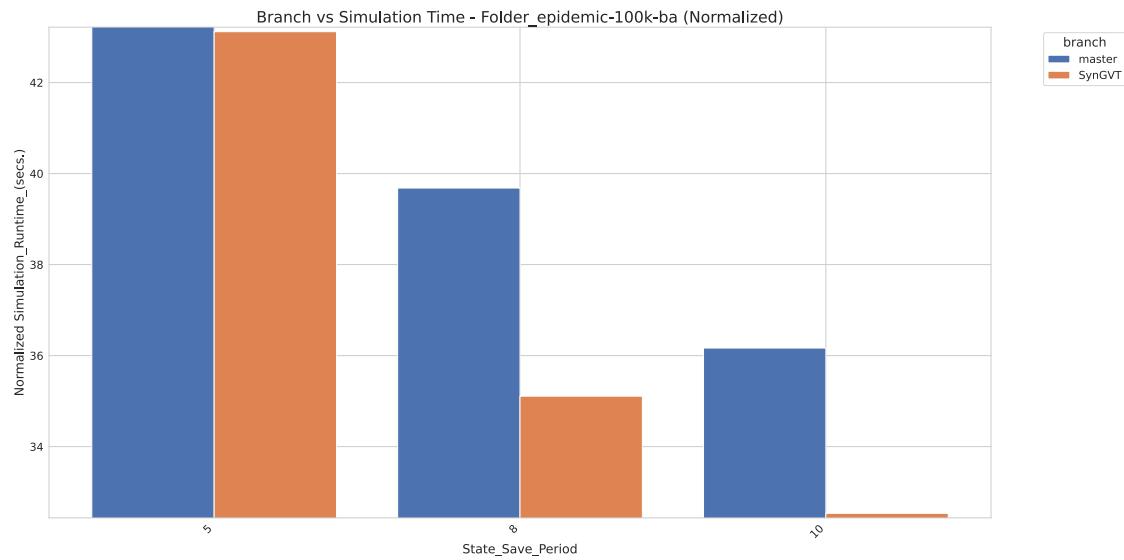


Figure 121: Branch vs Simulation Time_Folder.epidemic-100k-ba_normalized

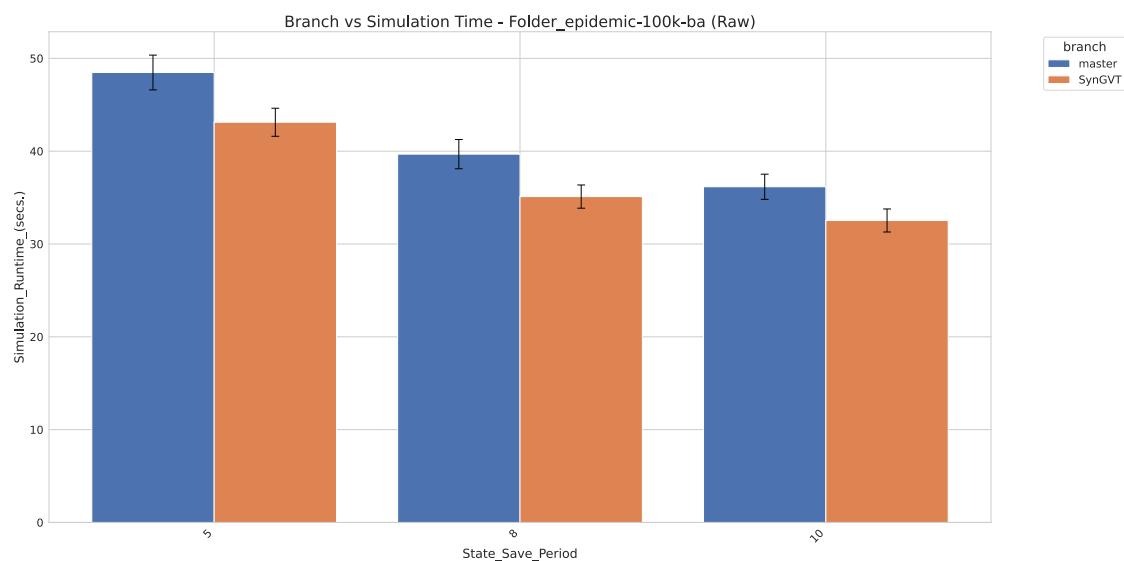


Figure 122: Branch vs Simulation Time_Folder.epidemic-100k-ba_raw

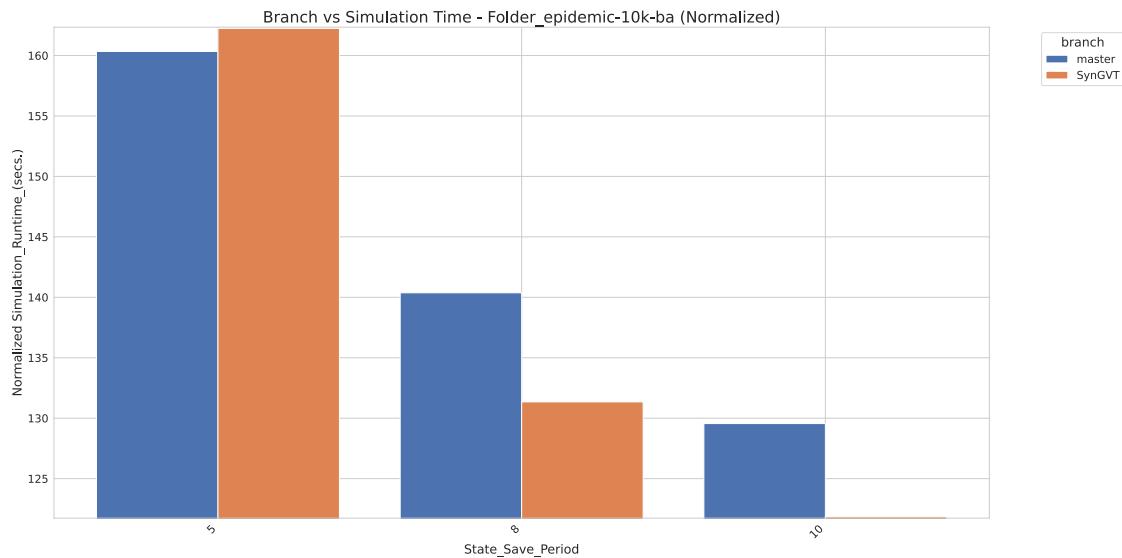


Figure 123: Branch vs Simulation Time_Folder_e pidemic-10k-ba_normalized

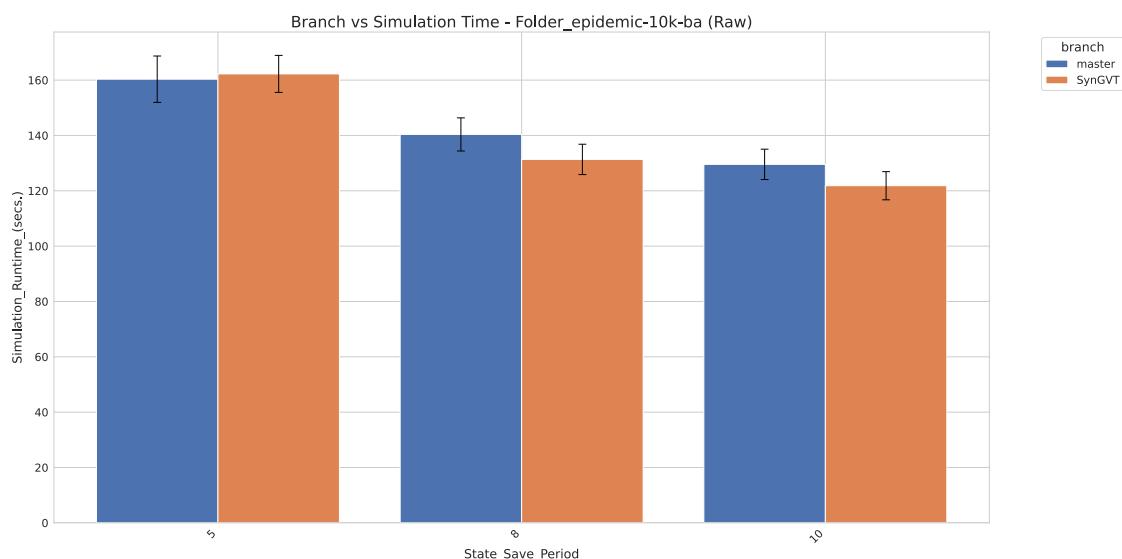


Figure 124: Branch vs Simulation Time_Folder_e pidemic-10k-ba.raw

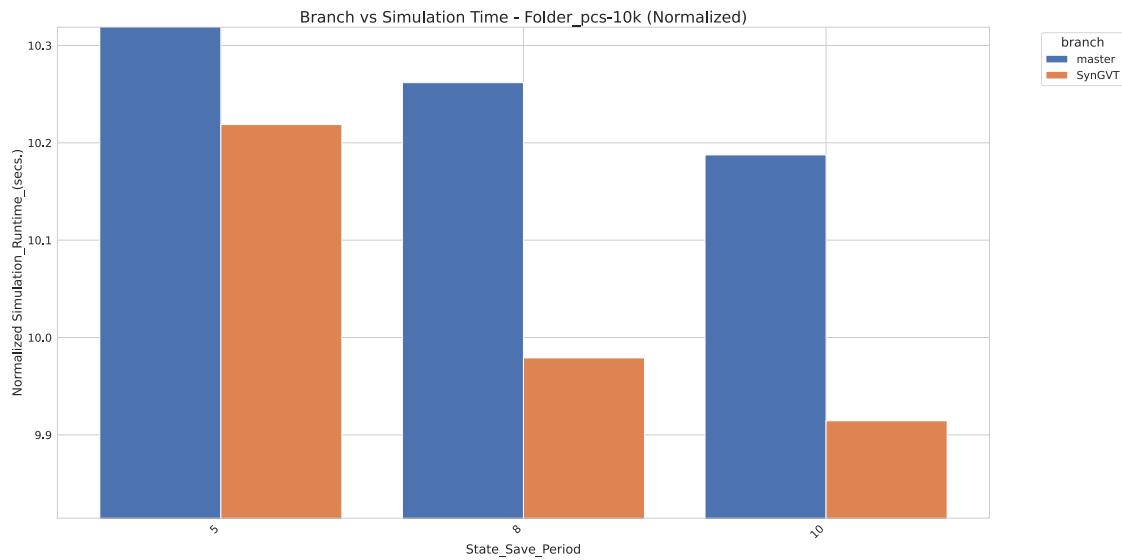


Figure 125: Branch vs Simulation Time_Folder_pcs-10k_normalized

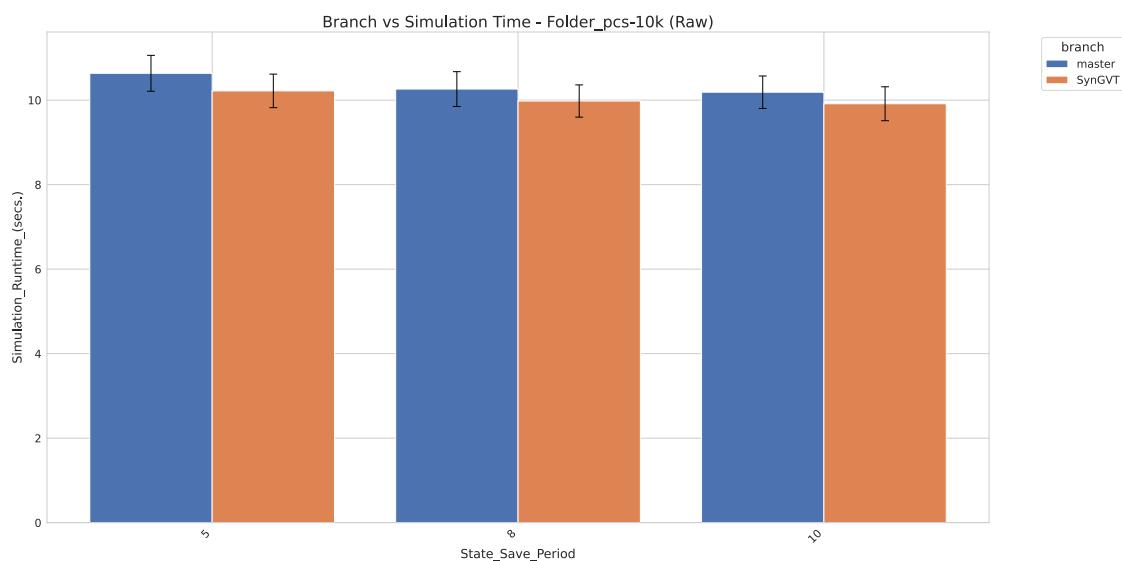


Figure 126: Branch vs Simulation Time_Folder_pcs-10k_raw

9 hashing

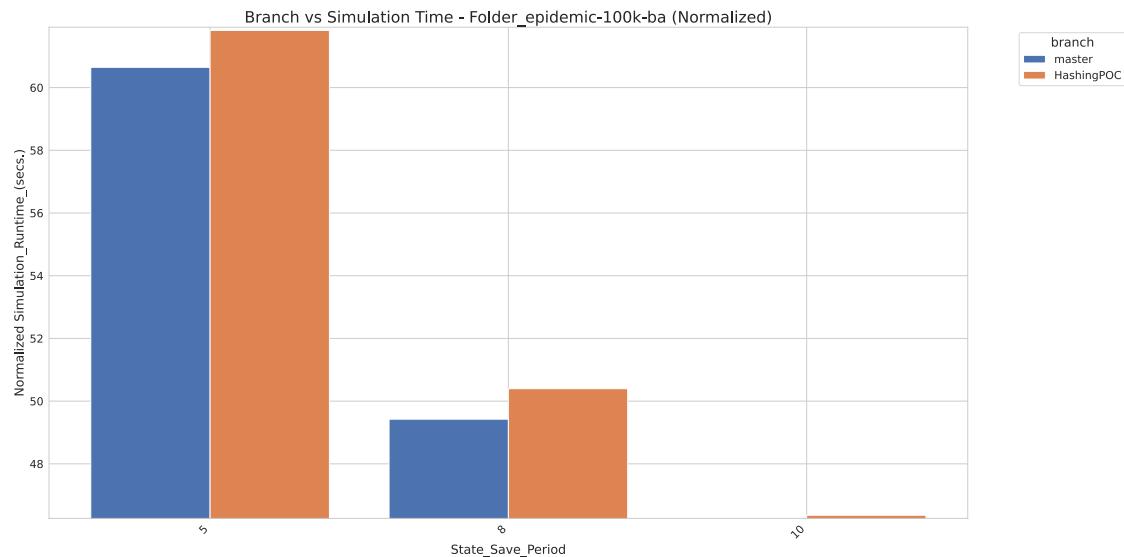


Figure 127: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

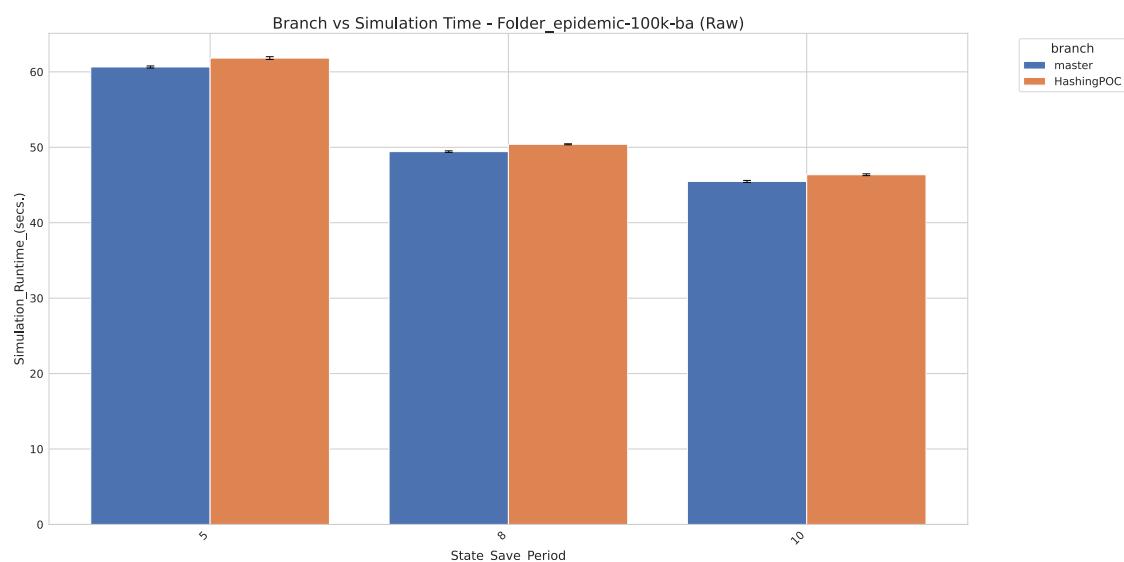


Figure 128: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

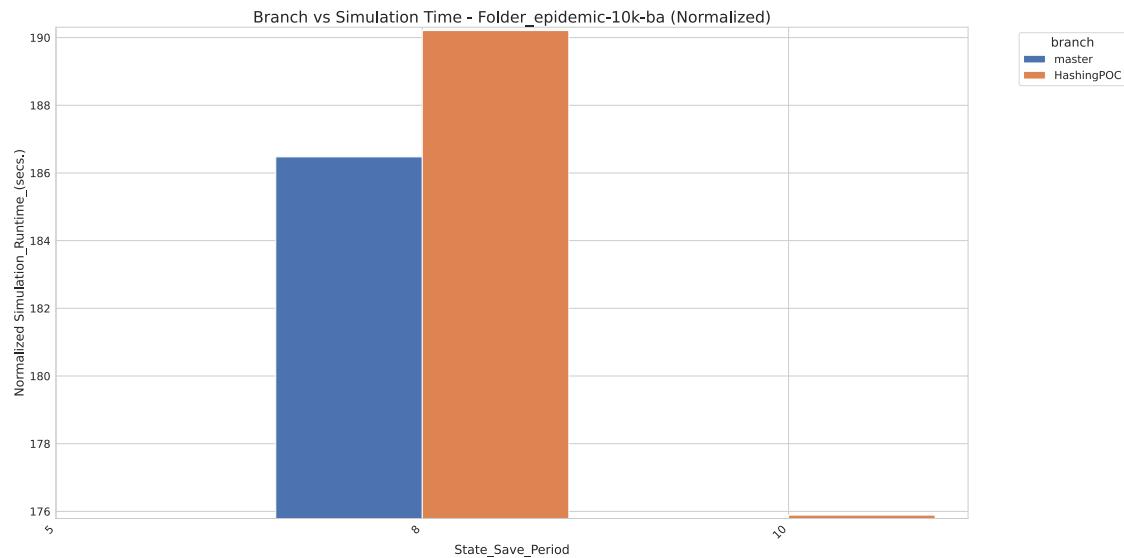


Figure 129: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

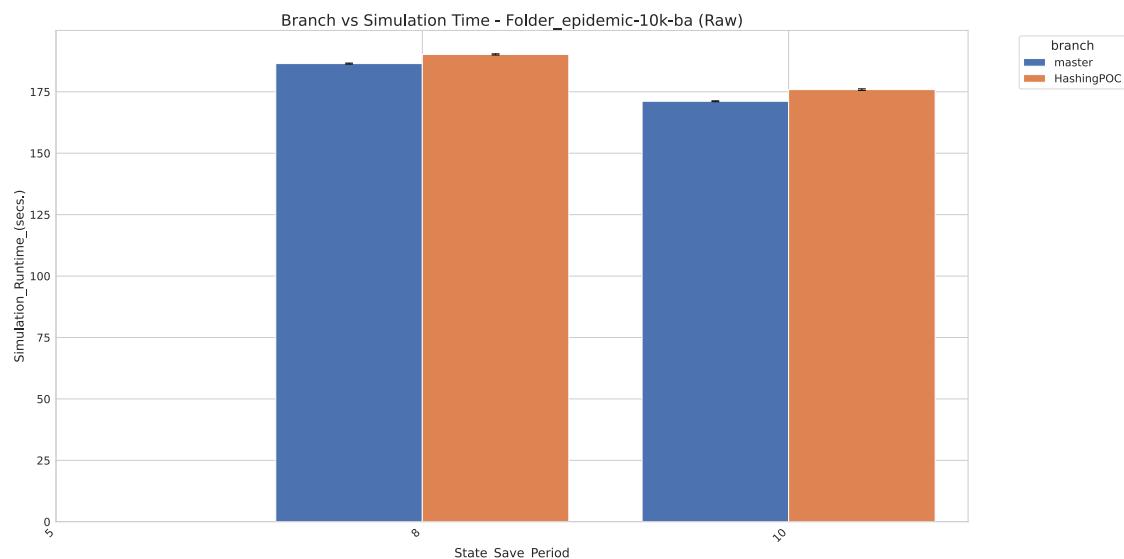


Figure 130: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

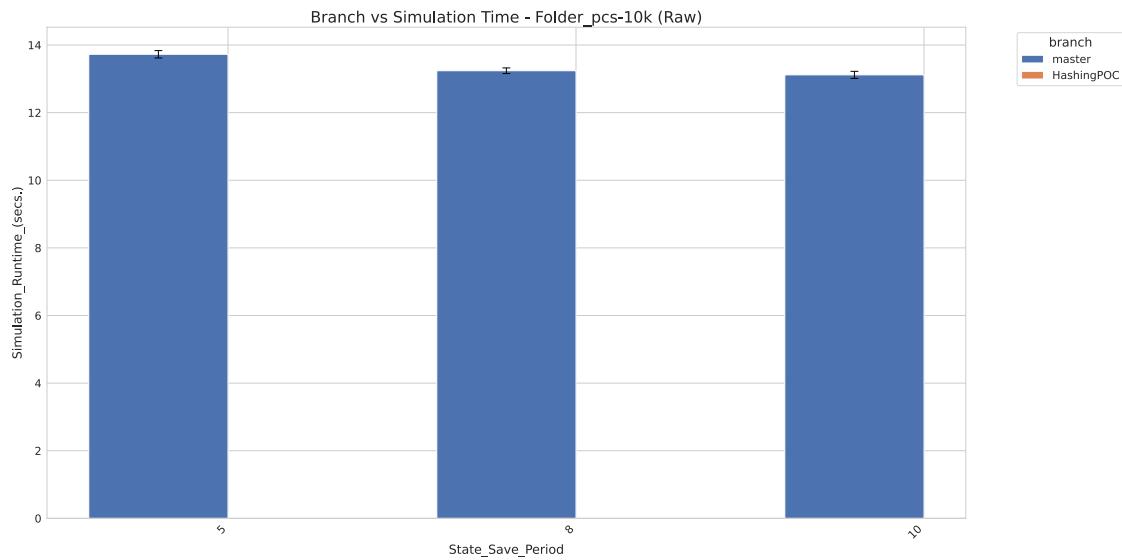


Figure 131: Branch vs Simulation Time_Folder_pcs-10k_raw

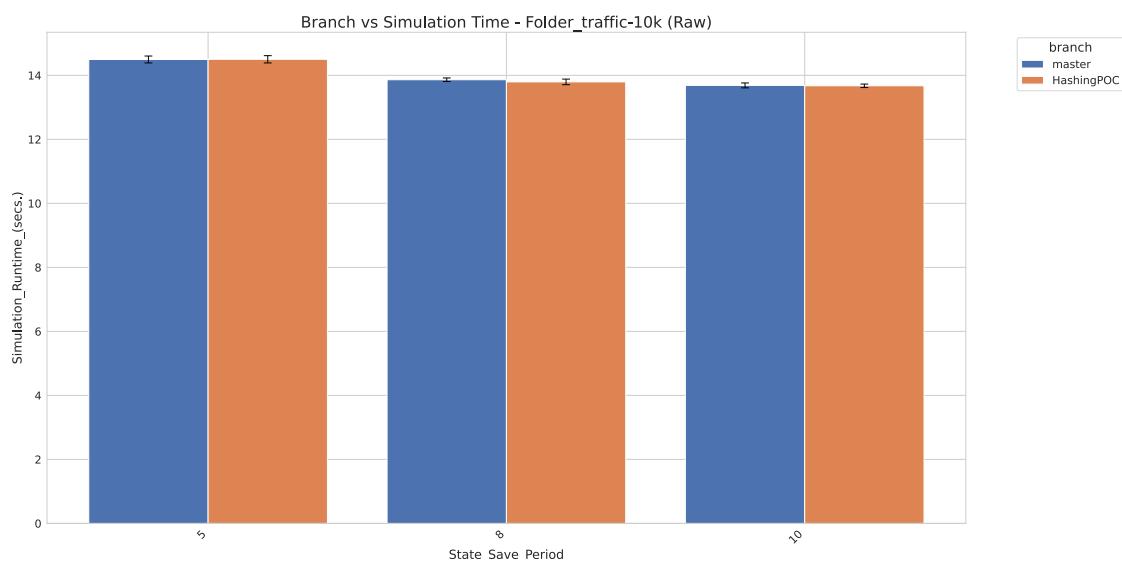


Figure 132: Branch vs Simulation Time_Folder_traffic-10k_raw

10 hashing_local

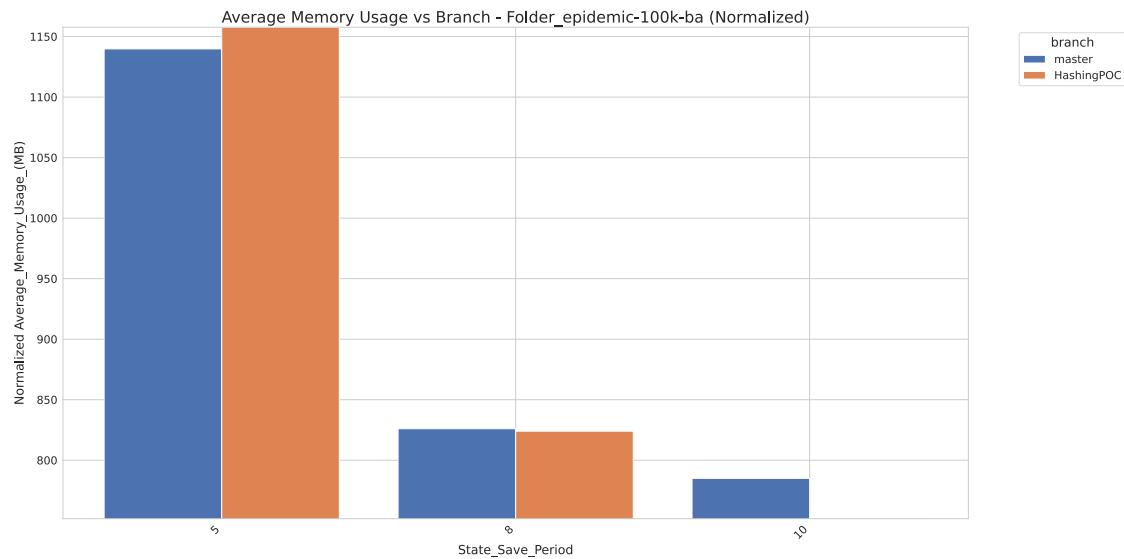


Figure 133: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

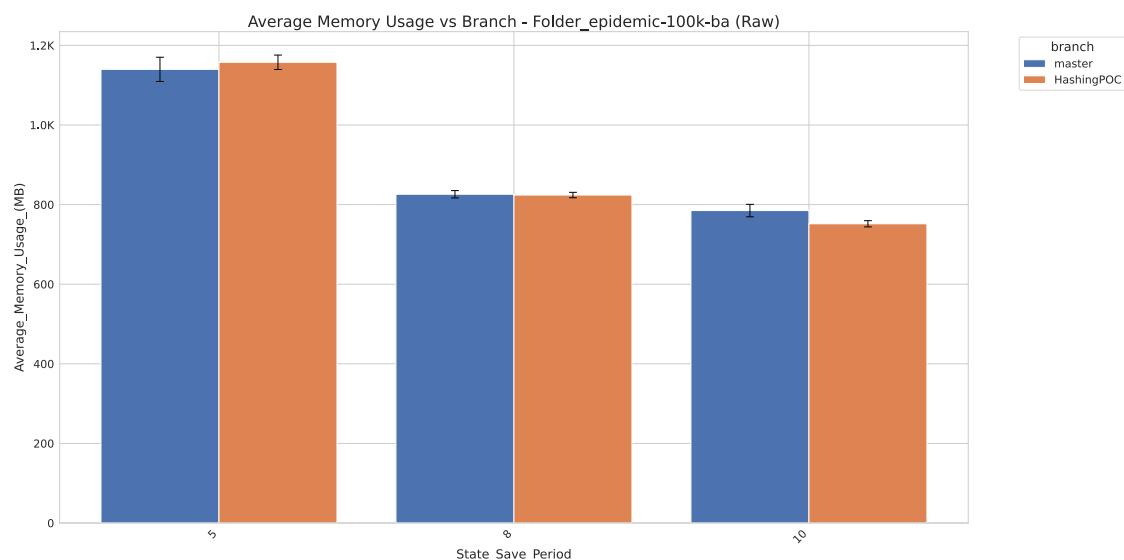


Figure 134: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

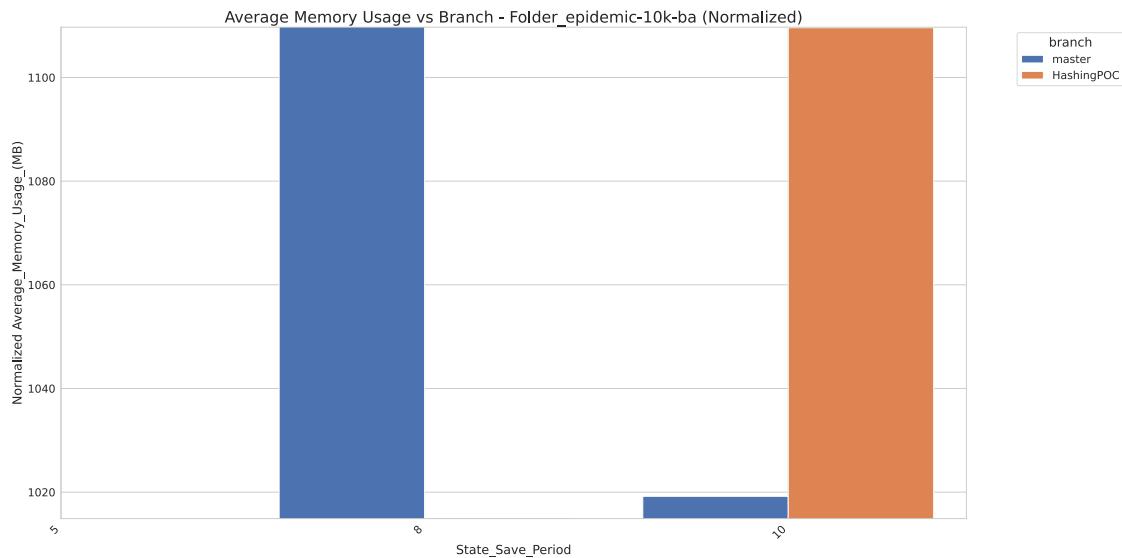


Figure 135: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_normalized

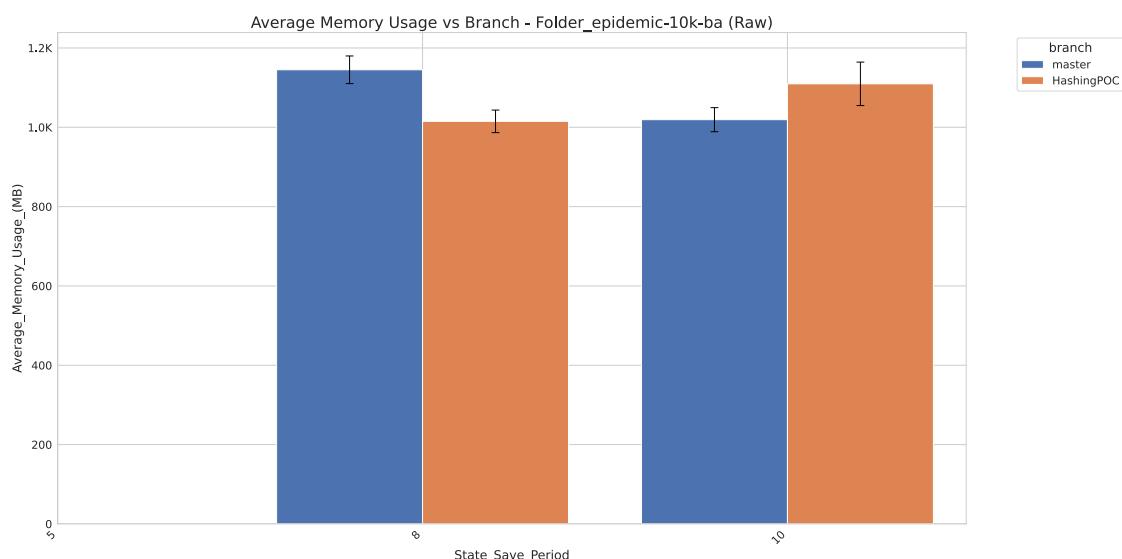


Figure 136: Average Memory Usage vs Branch_Folder_epidemic-10k-ba_raw

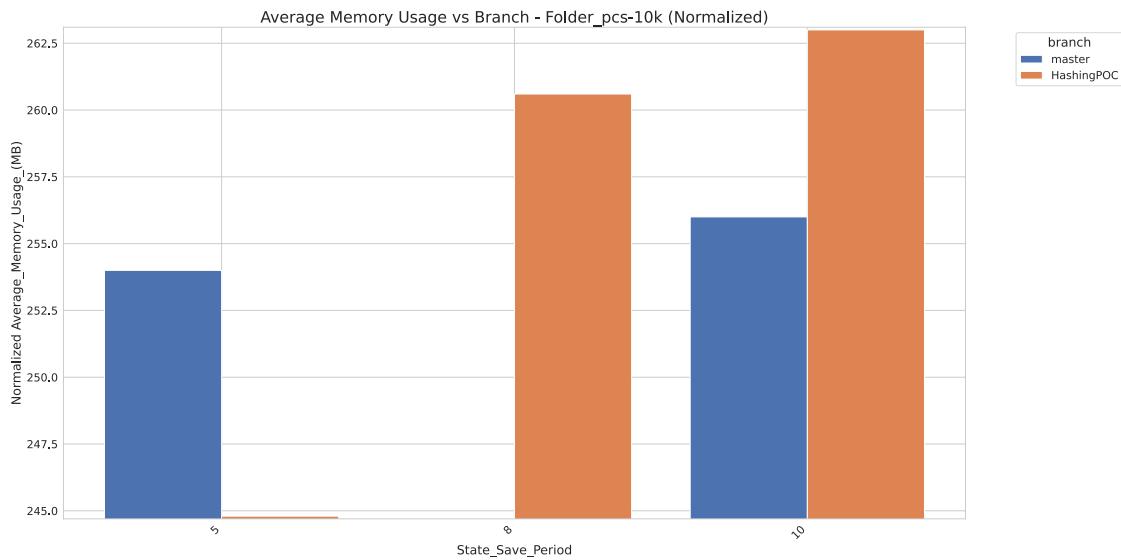


Figure 137: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

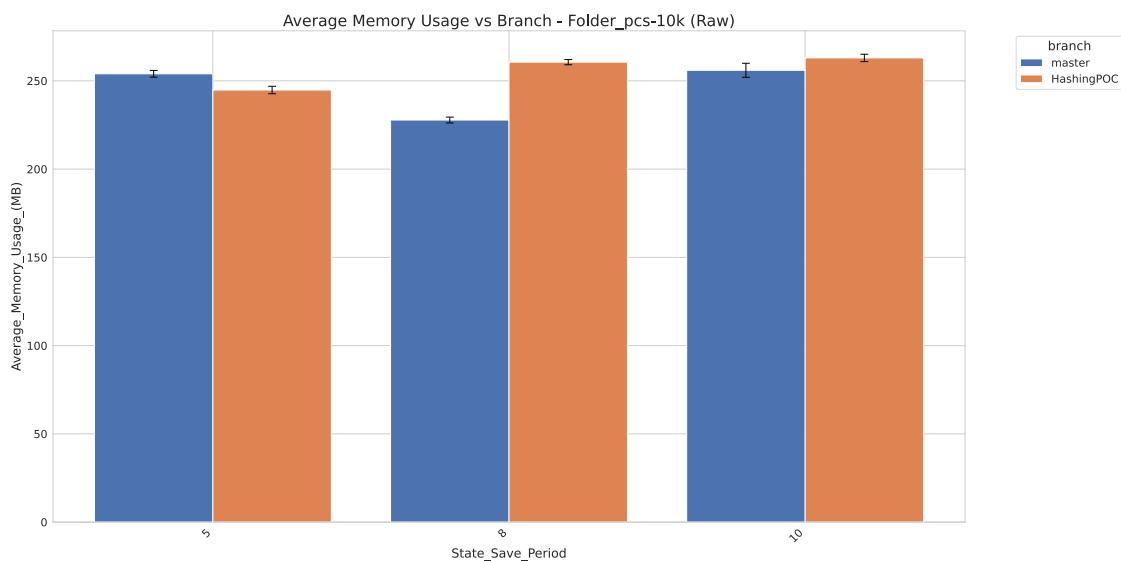


Figure 138: Average Memory Usage vs Branch_Folder_pcs-10k_raw

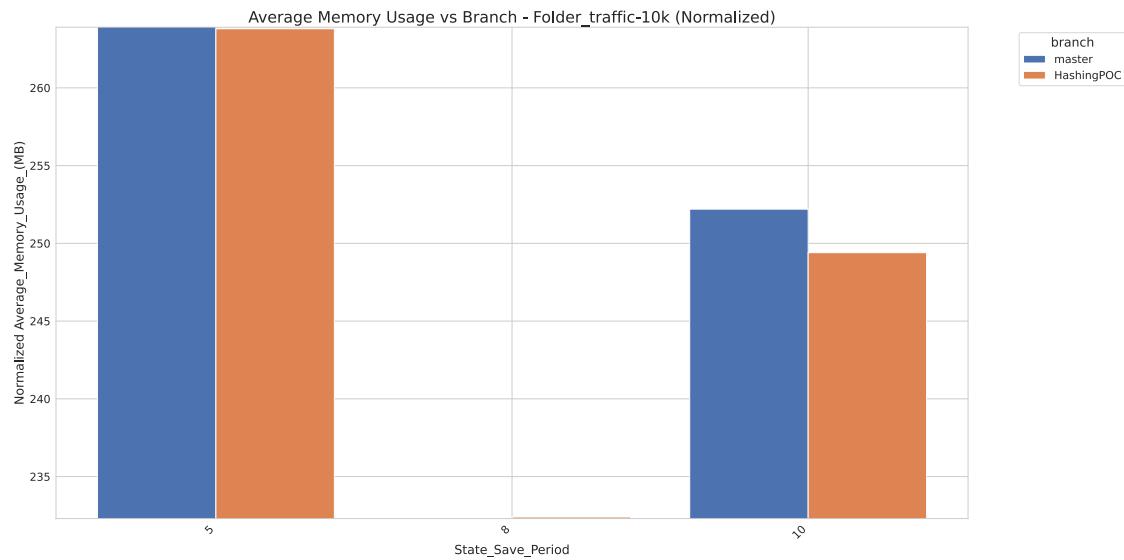


Figure 139: Average Memory Usage vs Branch_Folder_traffic-10k_normalized

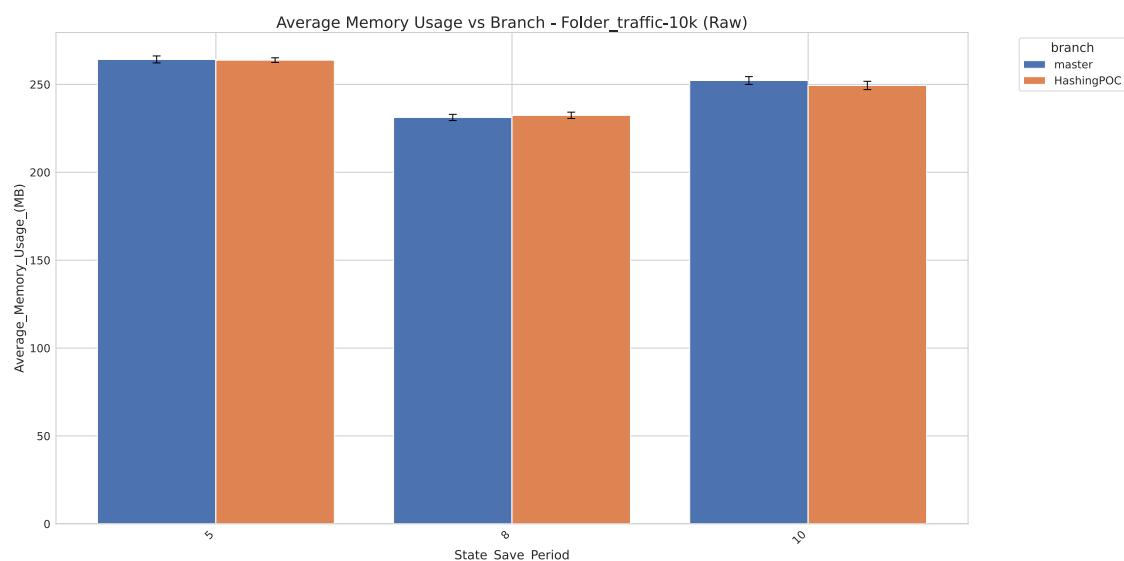


Figure 140: Average Memory Usage vs Branch_Folder_traffic-10k_raw

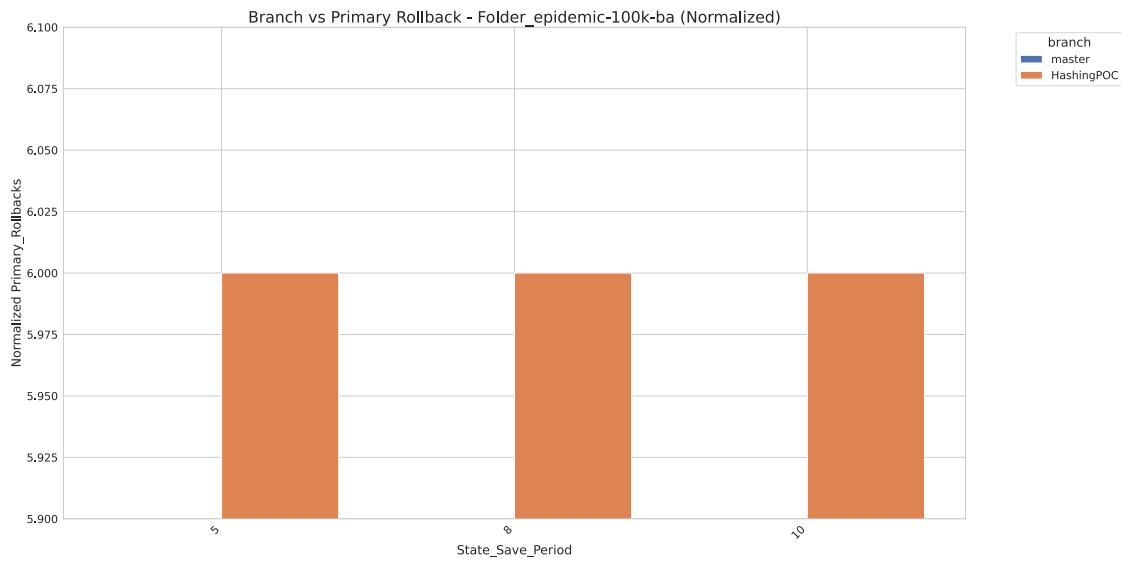


Figure 141: Branch vs Primary Rollback_Folder_epidemic-100k-ba_normalized

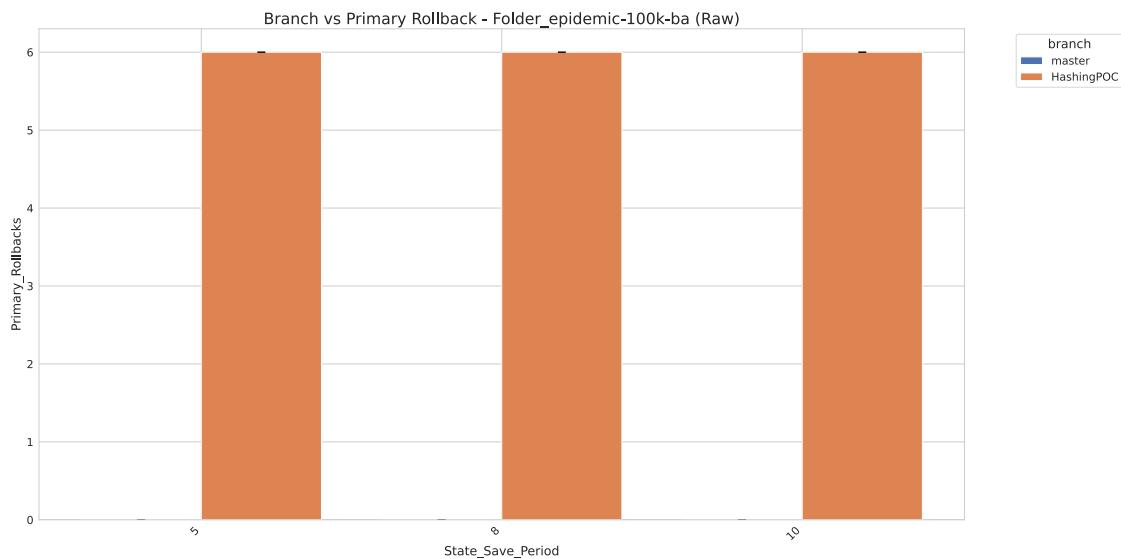


Figure 142: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

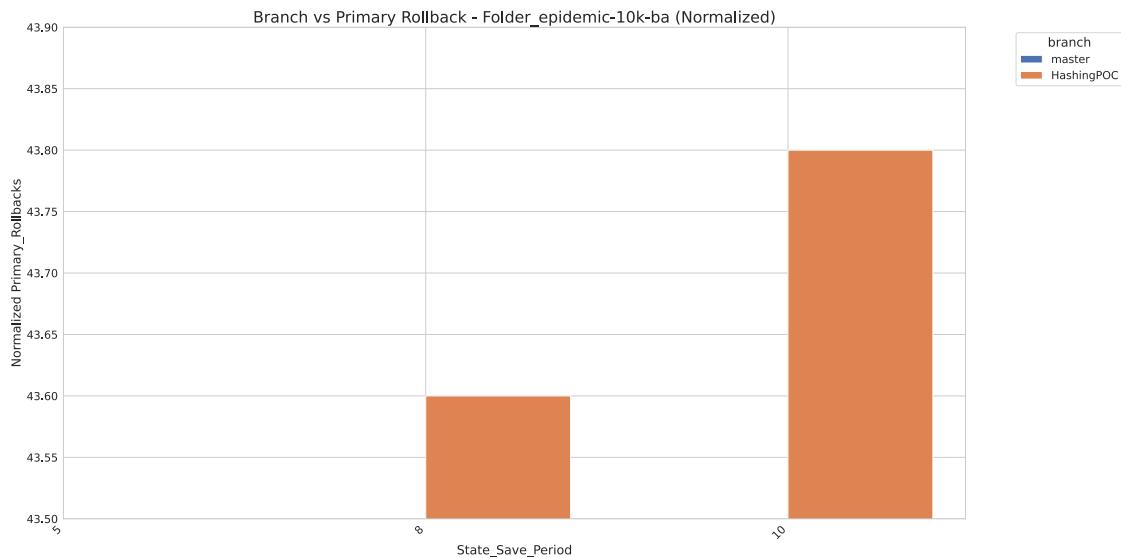


Figure 143: Branch vs Primary Rollback_Folder.epidemic-10k-ba_normalized

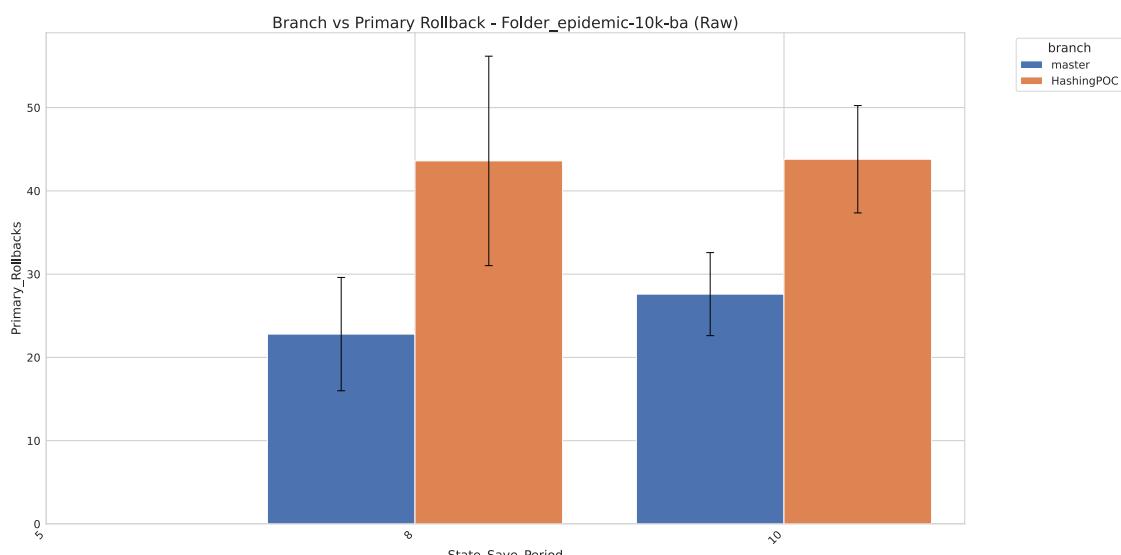


Figure 144: Branch vs Primary Rollback_Folder.epidemic-10k-ba_raw

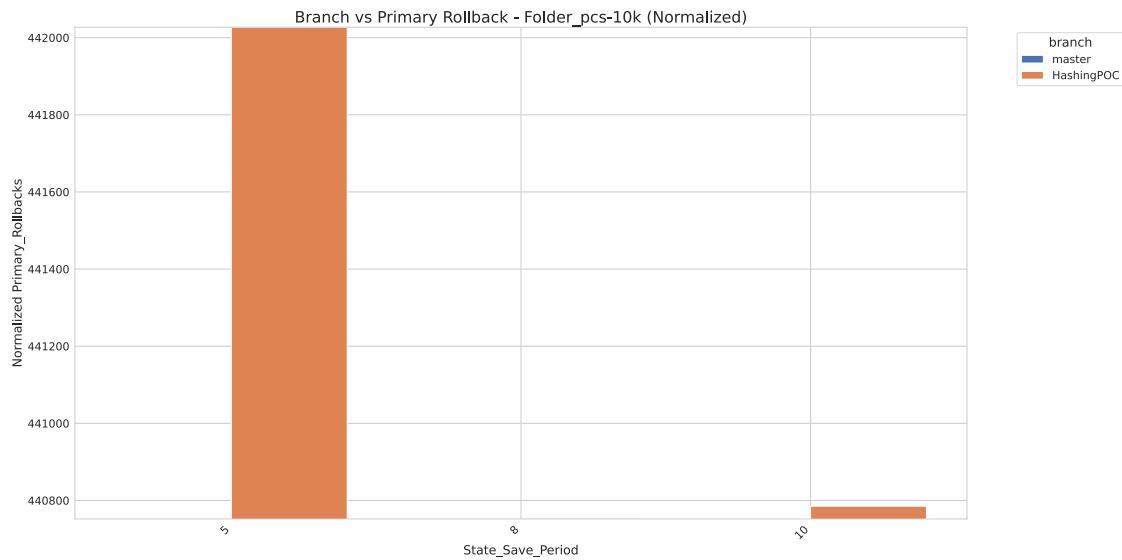


Figure 145: Branch vs Primary Rollback_Folder_pcs-10k_normalized

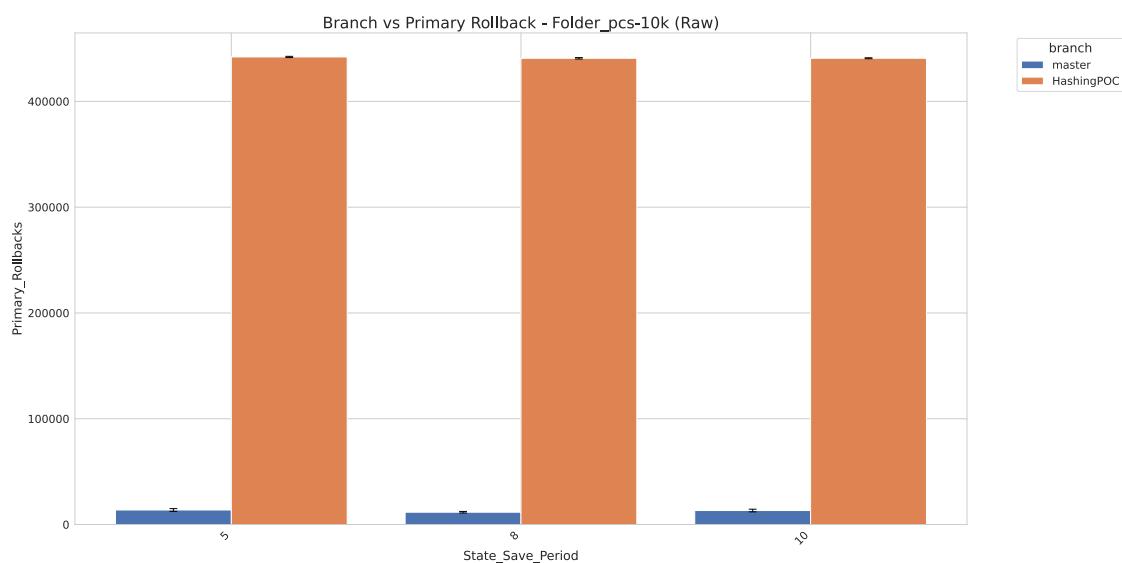


Figure 146: Branch vs Primary Rollback_Folder_pcs-10k_raw

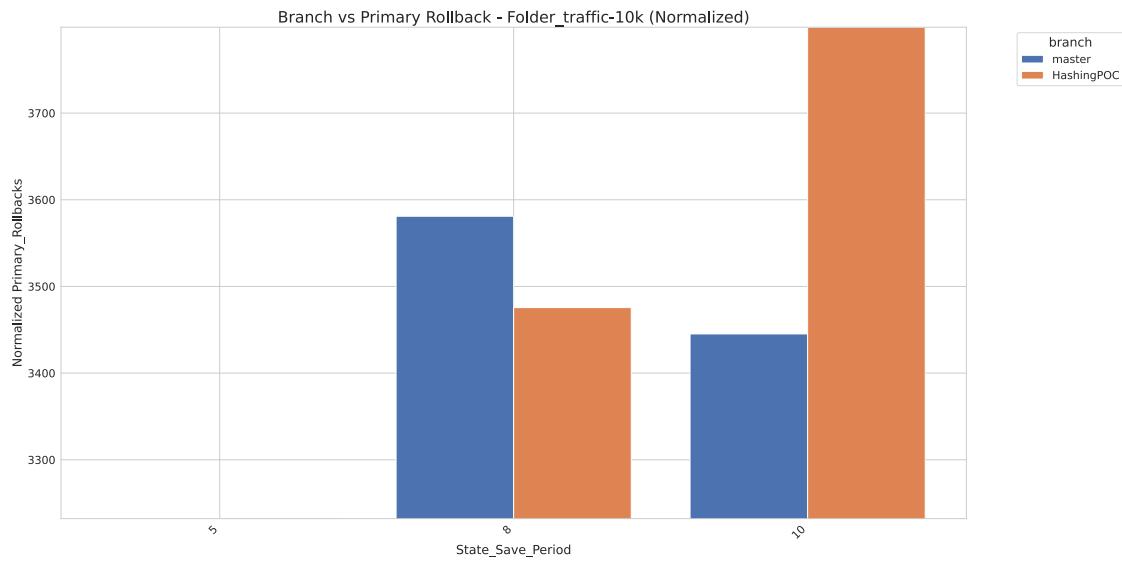


Figure 147: Branch vs Primary Rollback_Folder_traffic-10k_normalized

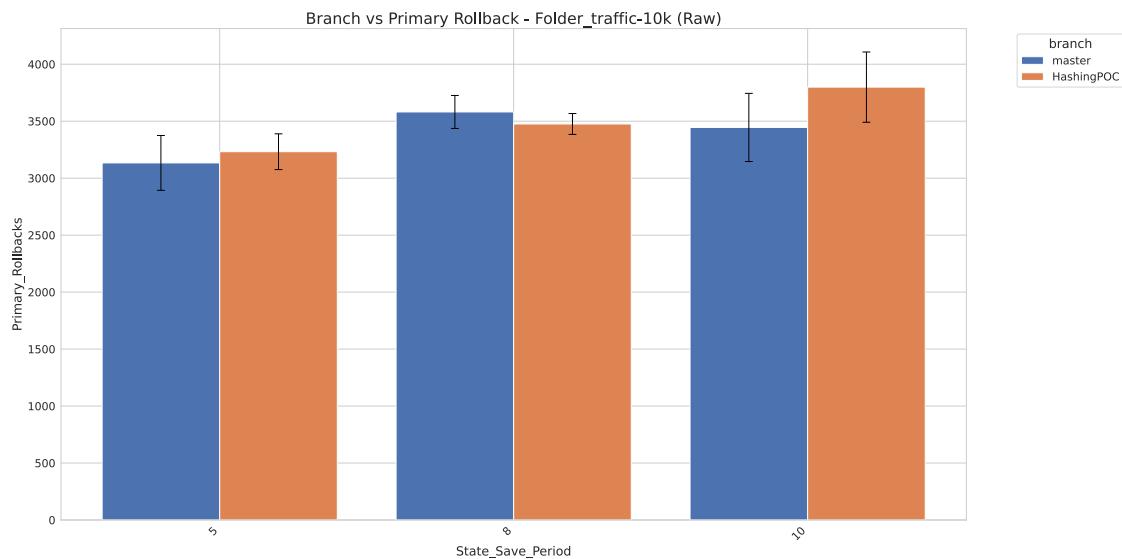


Figure 148: Branch vs Primary Rollback_Folder_traffic-10k_raw

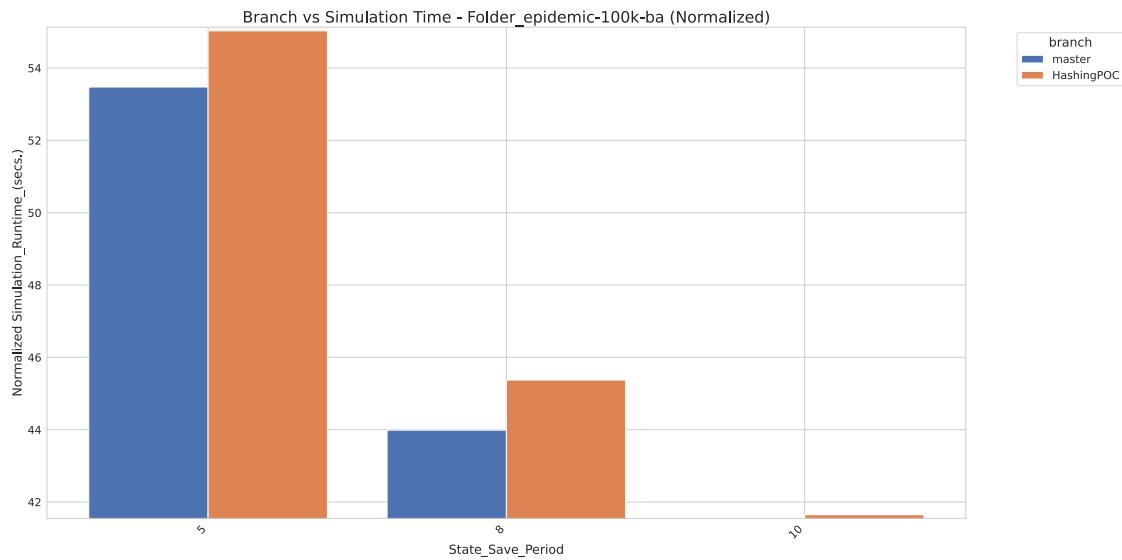


Figure 149: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

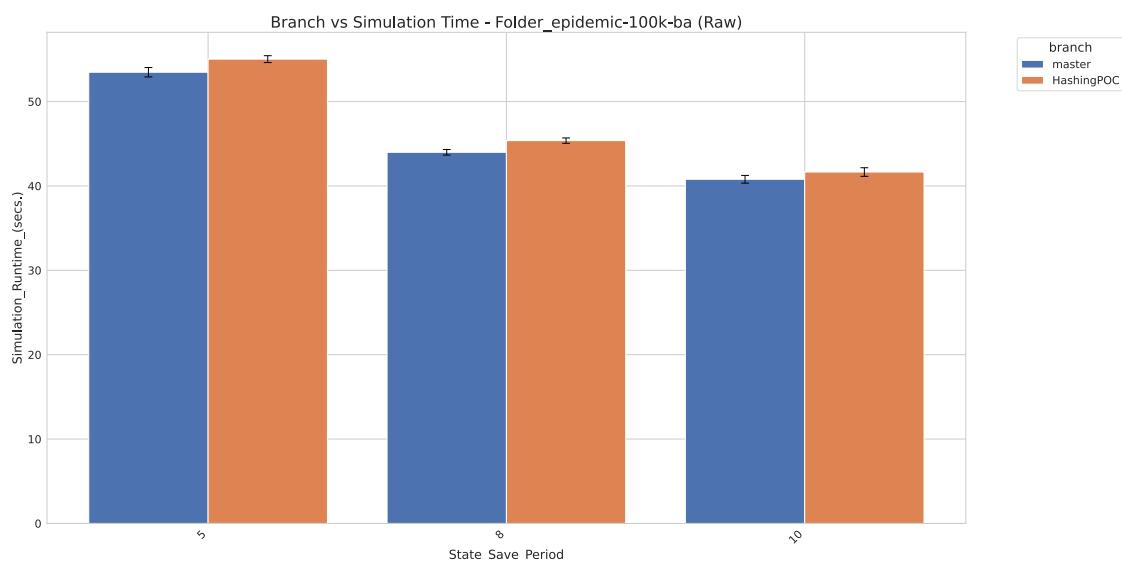


Figure 150: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

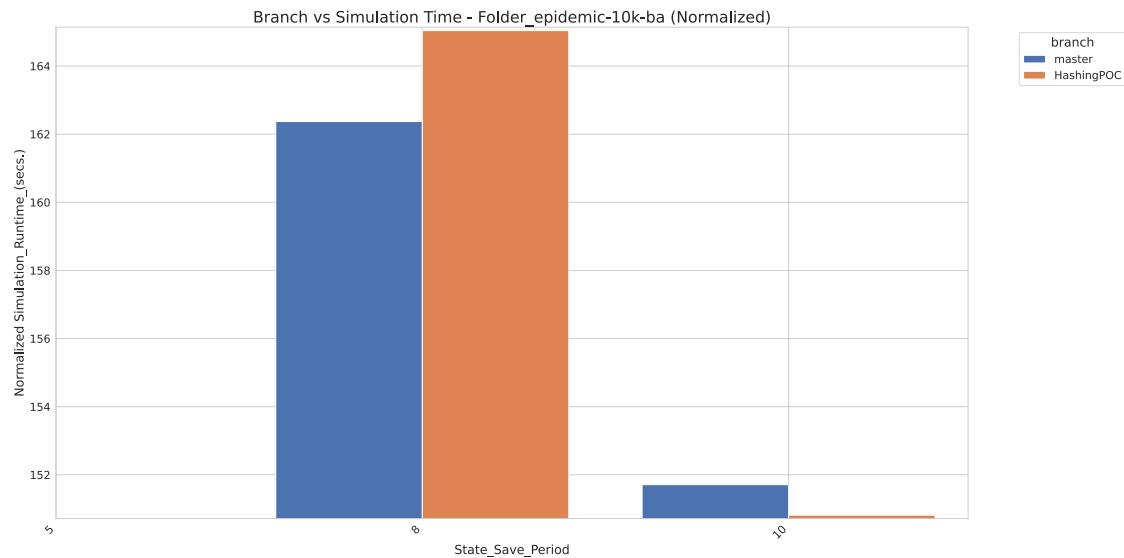


Figure 151: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

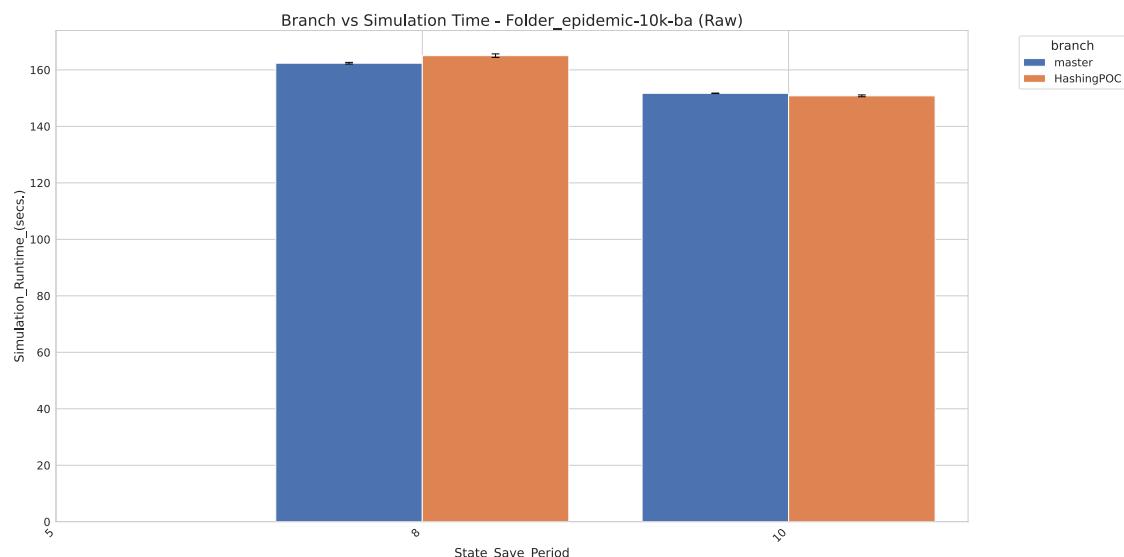


Figure 152: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

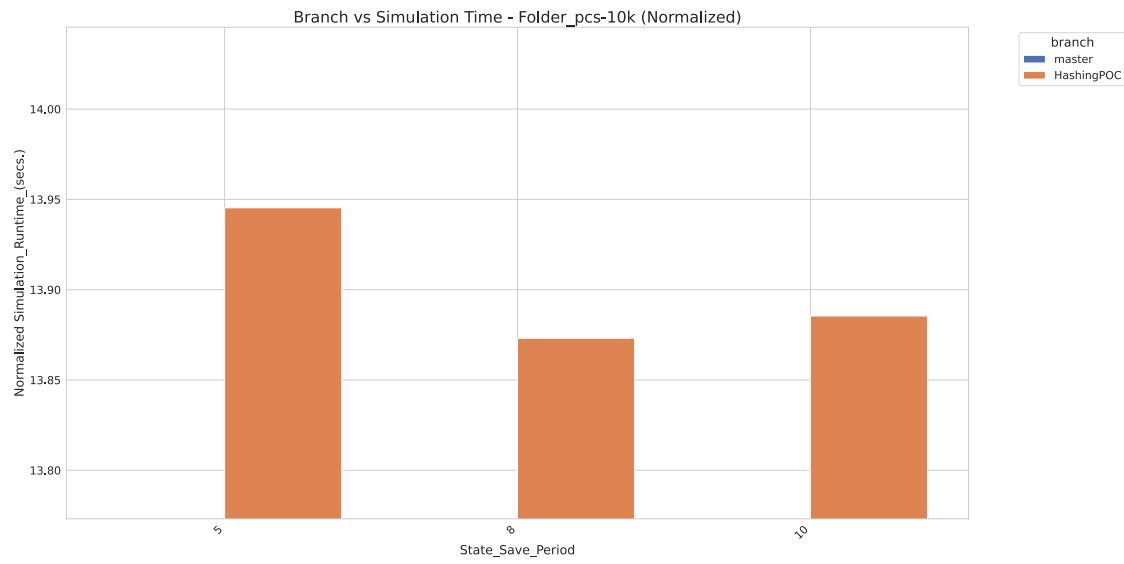


Figure 153: Branch vs Simulation Time_Folder_pcs-10k_normalized

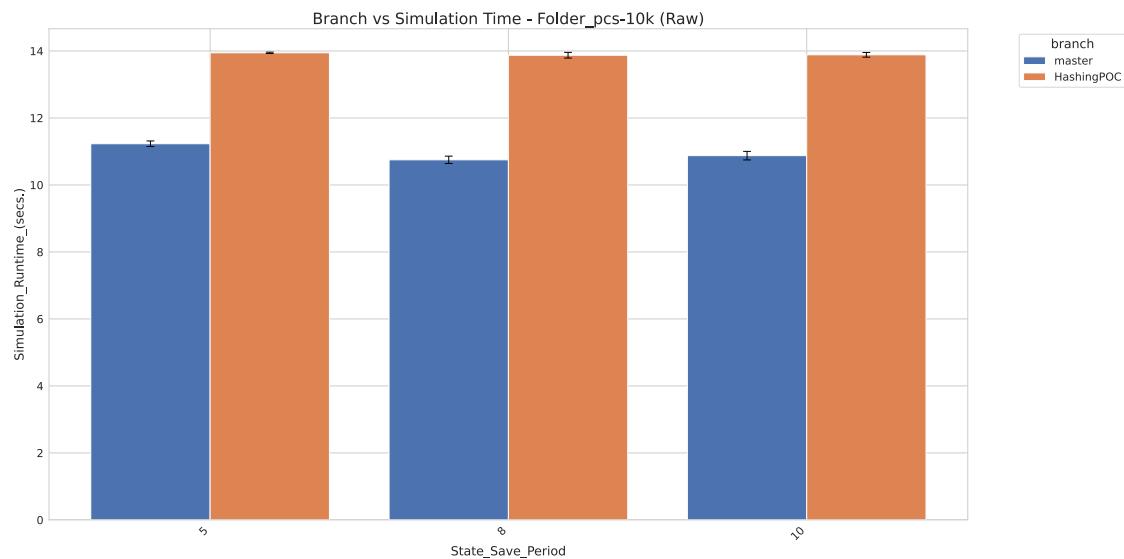


Figure 154: Branch vs Simulation Time_Folder_pcs-10k_raw

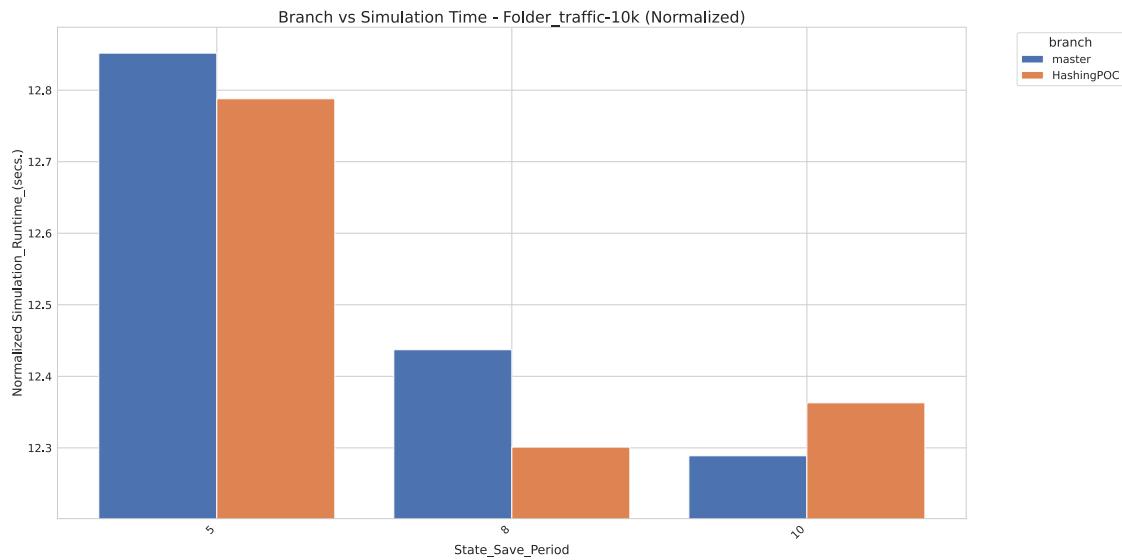


Figure 155: Branch vs Simulation Time_Folder_traffic-10k_normalized

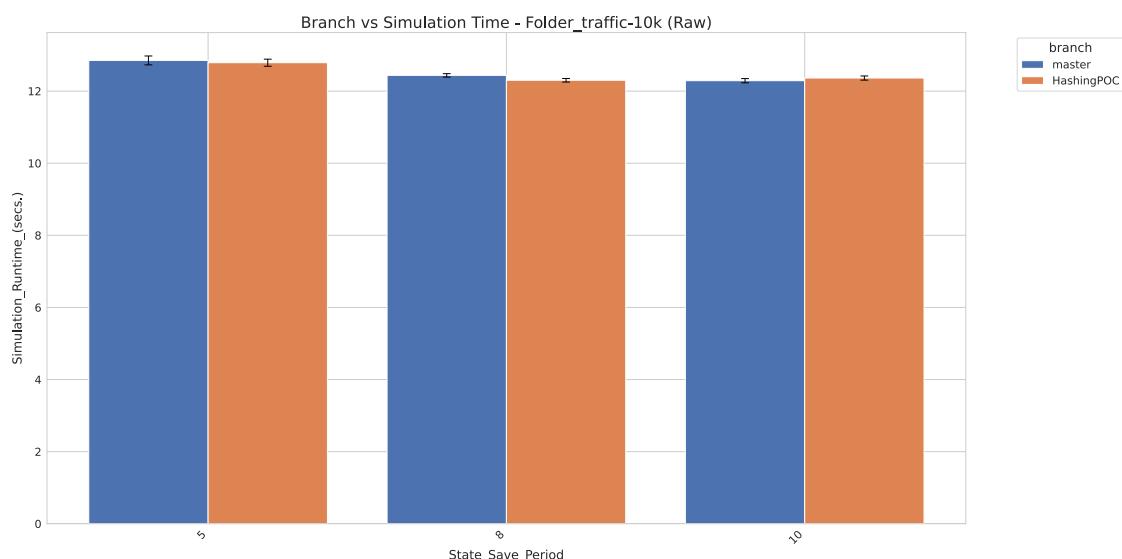


Figure 156: Branch vs Simulation Time_Folder_traffic-10k_raw

11 SIMD

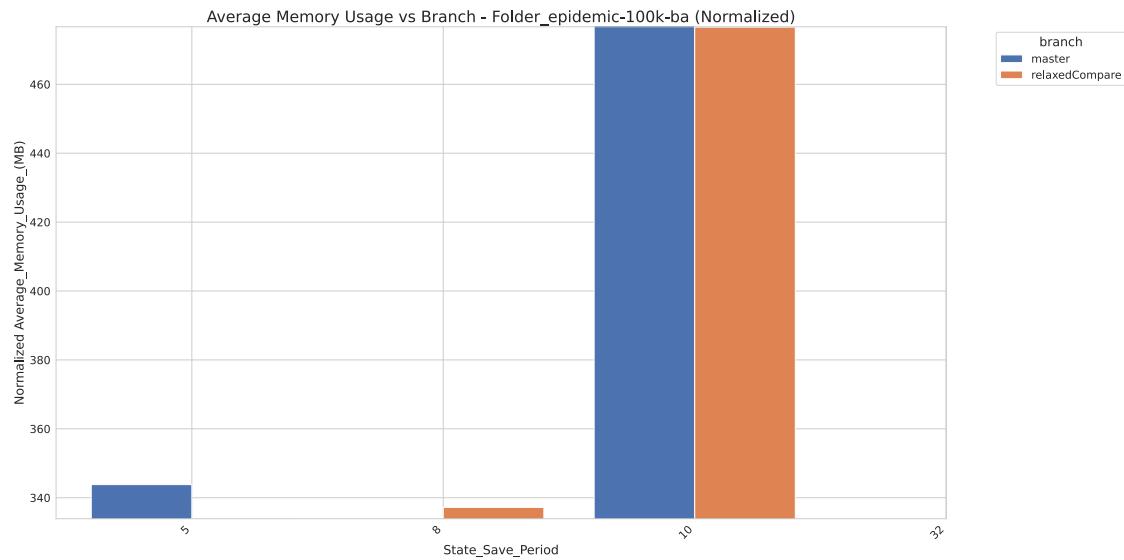


Figure 157: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

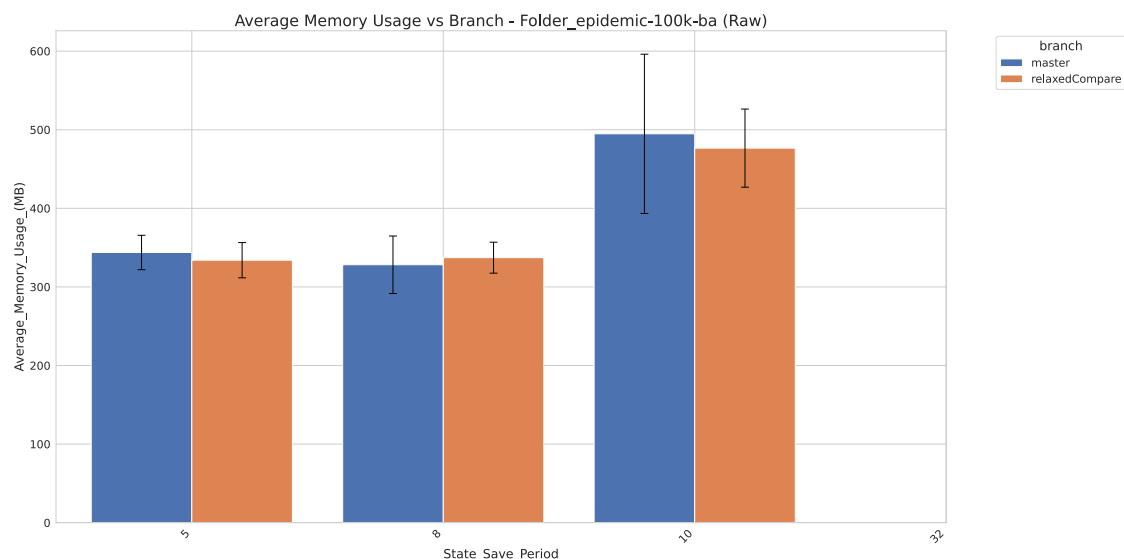


Figure 158: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

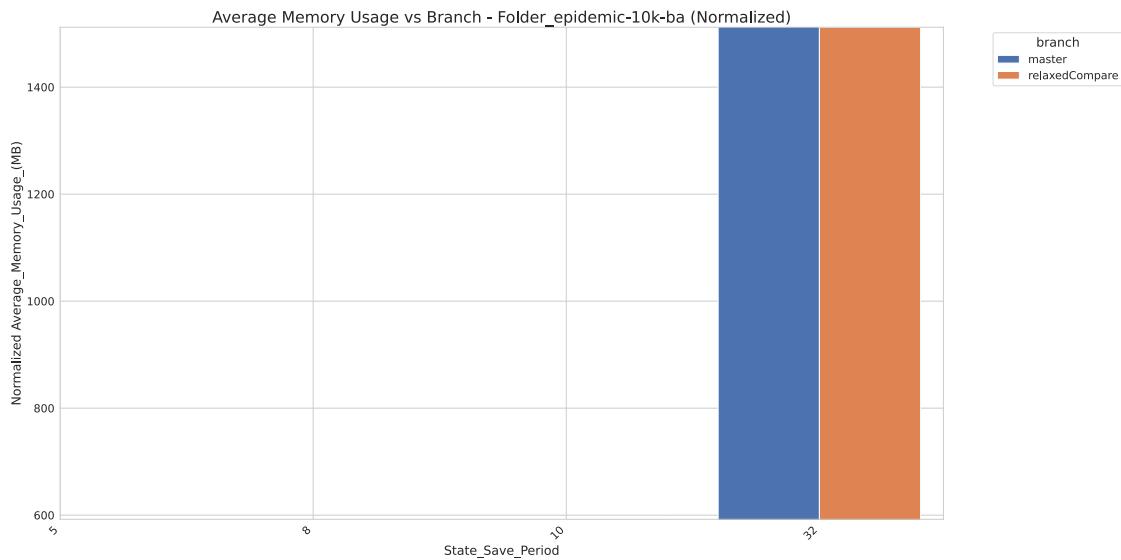


Figure 159: Average Memory Usage vs Branch_Folder.epidemic-10k-ba_normalized

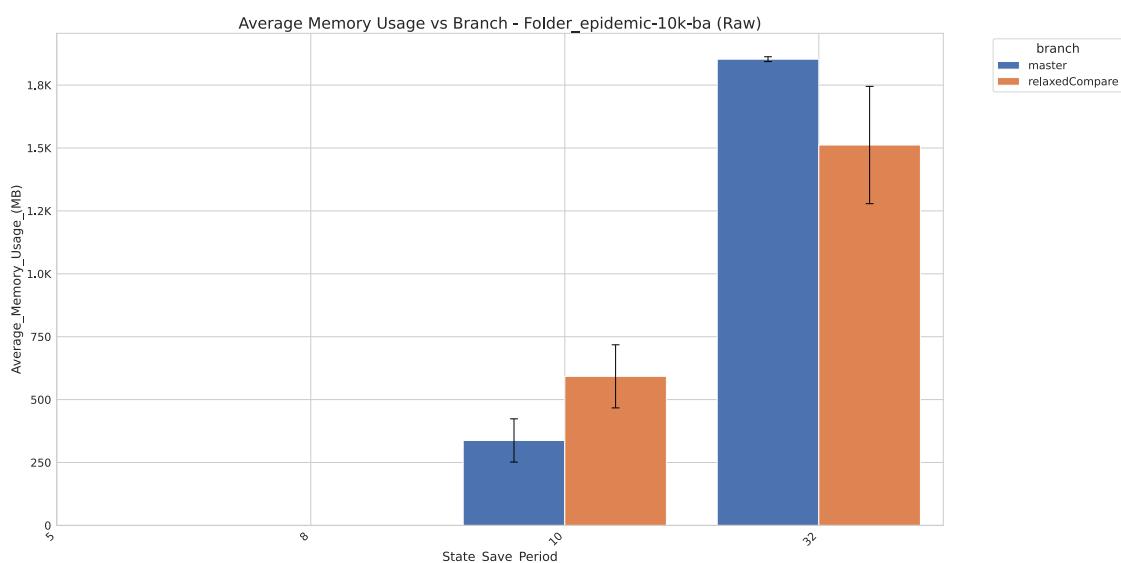


Figure 160: Average Memory Usage vs Branch_Folder.epidemic-10k-ba_raw

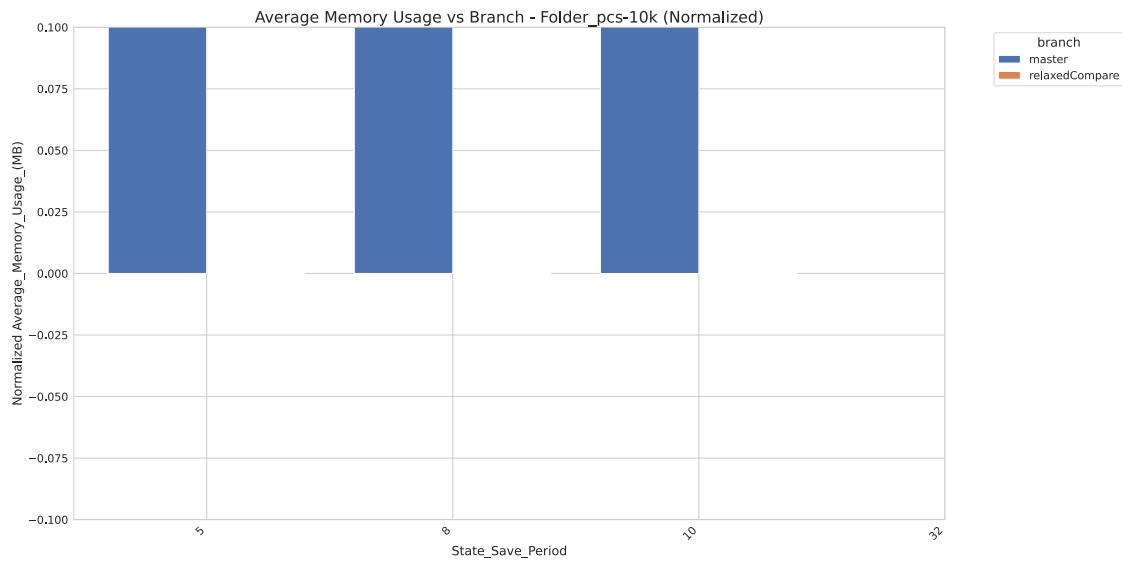


Figure 161: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

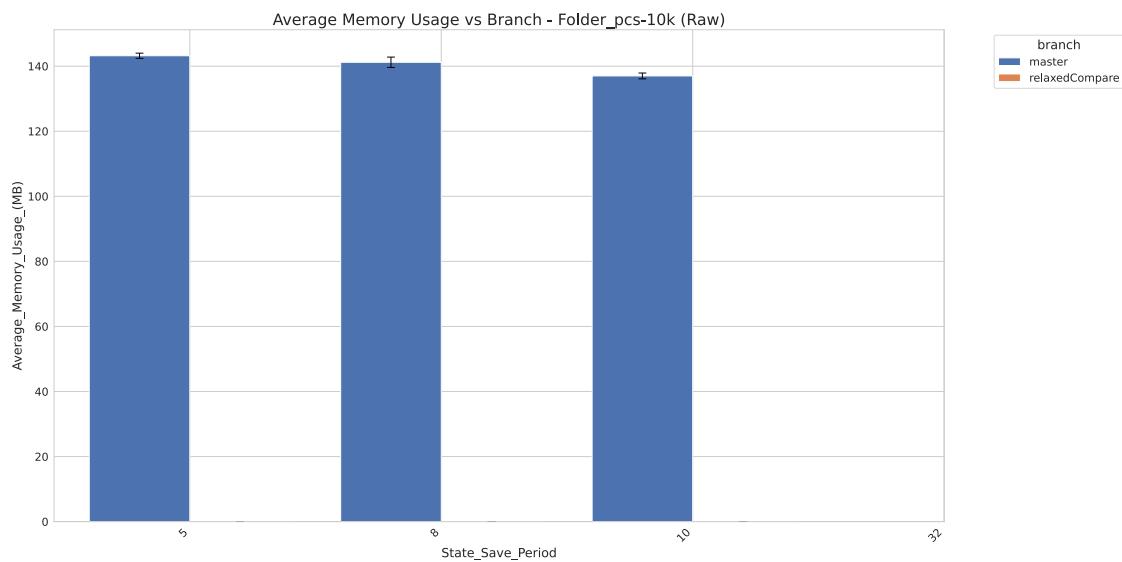


Figure 162: Average Memory Usage vs Branch_Folder_pcs-10k_raw

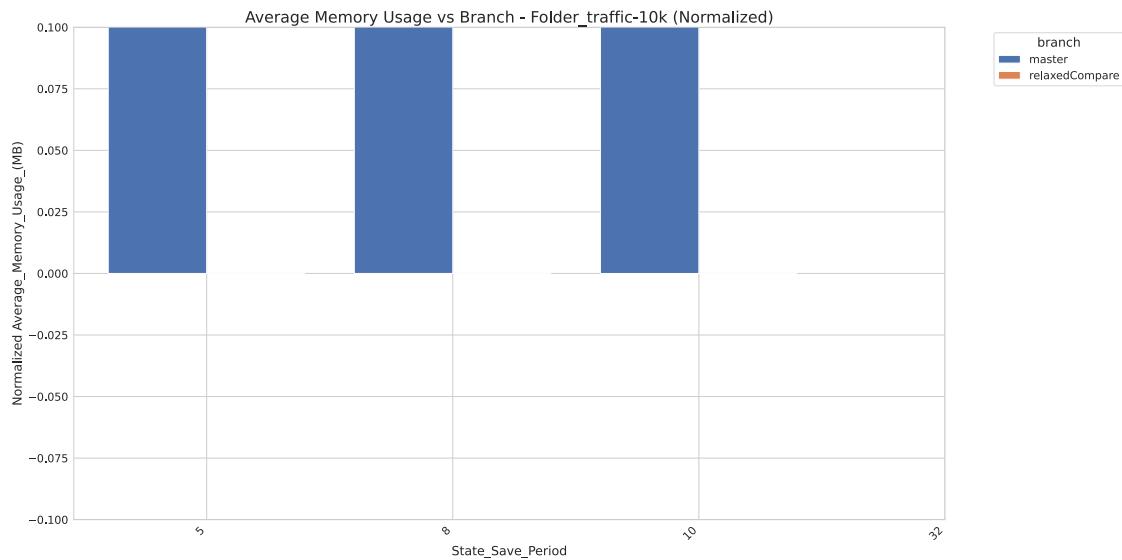


Figure 163: Average Memory Usage vs Branch_Folder_traffic-10k_normalized

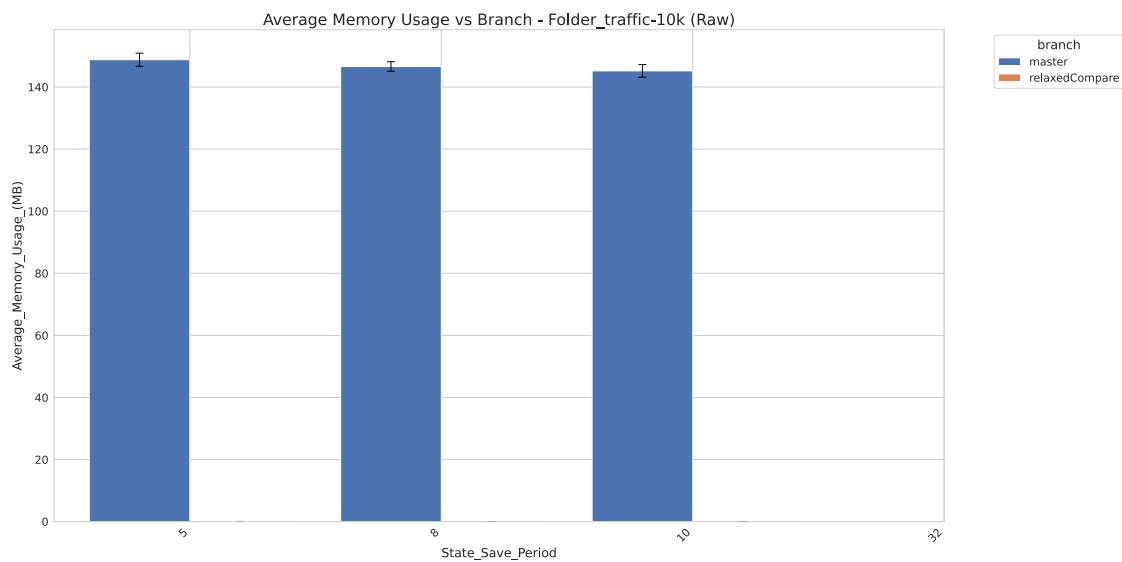


Figure 164: Average Memory Usage vs Branch_Folder_traffic-10k_raw

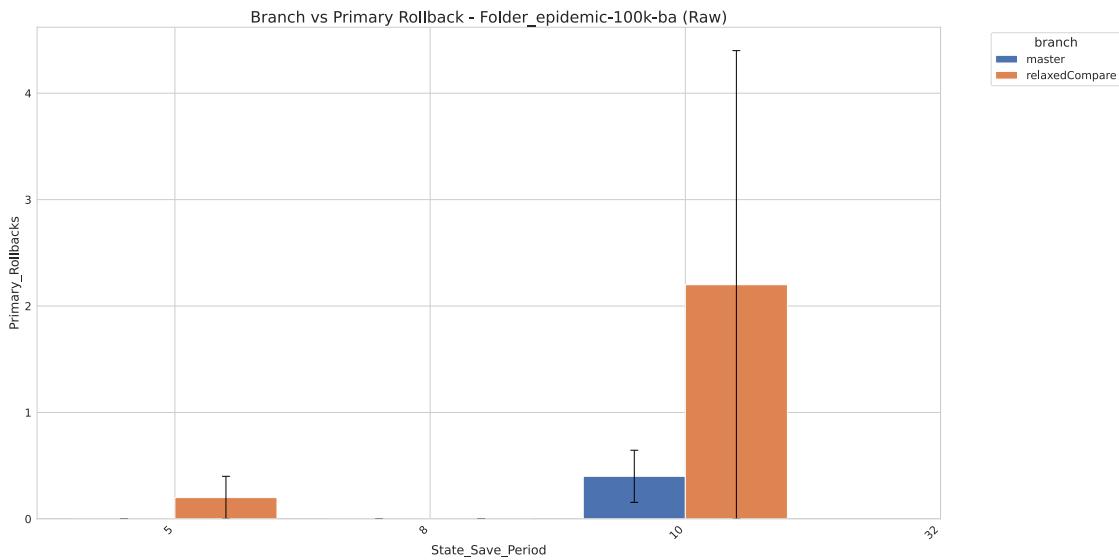


Figure 165: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

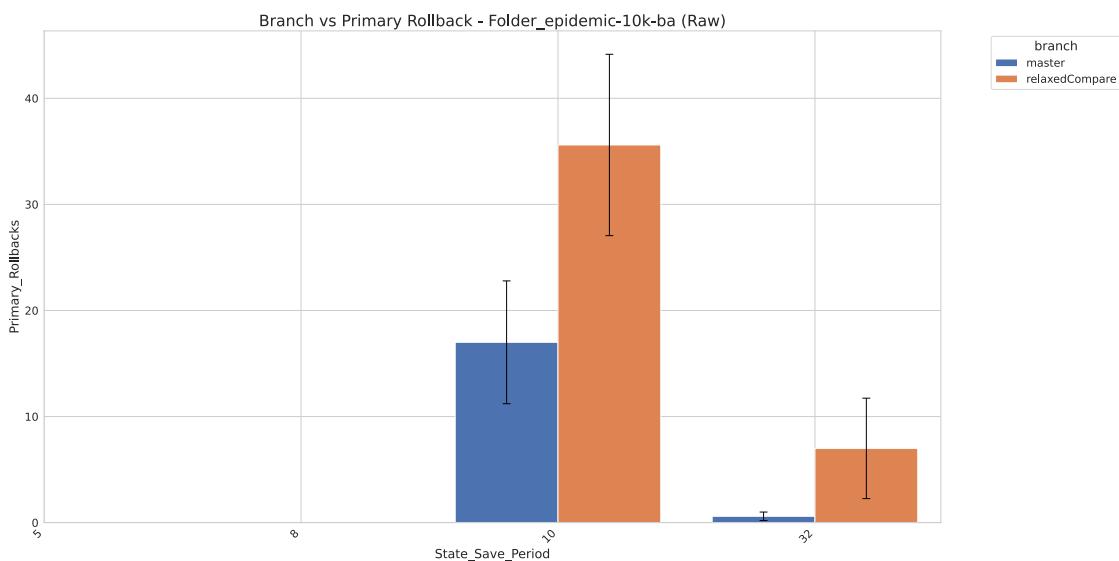


Figure 166: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

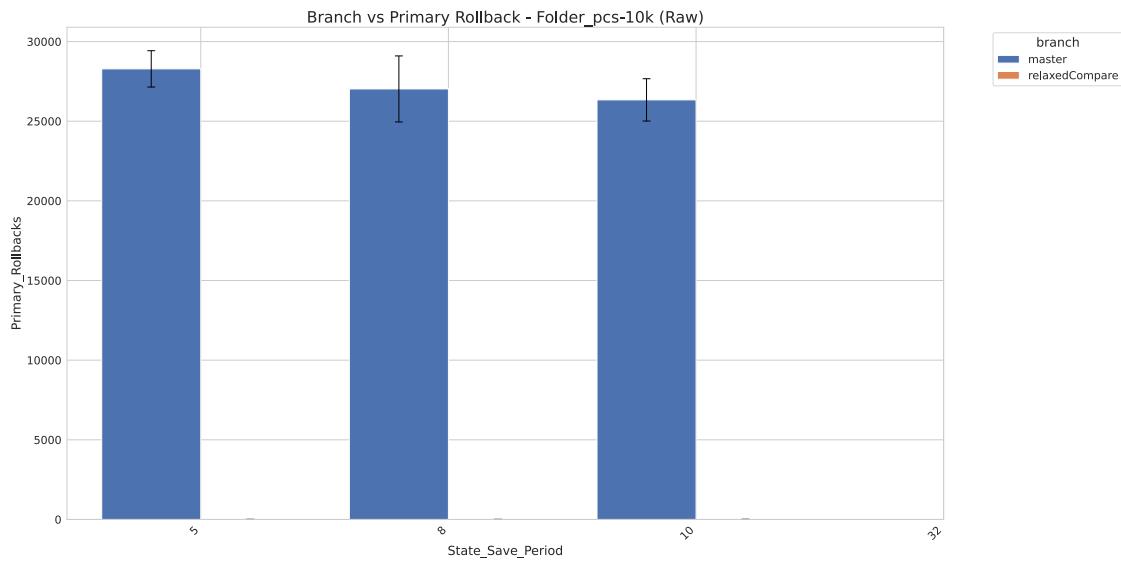


Figure 167: Branch vs Primary Rollback_Folder_pcs-10k_raw

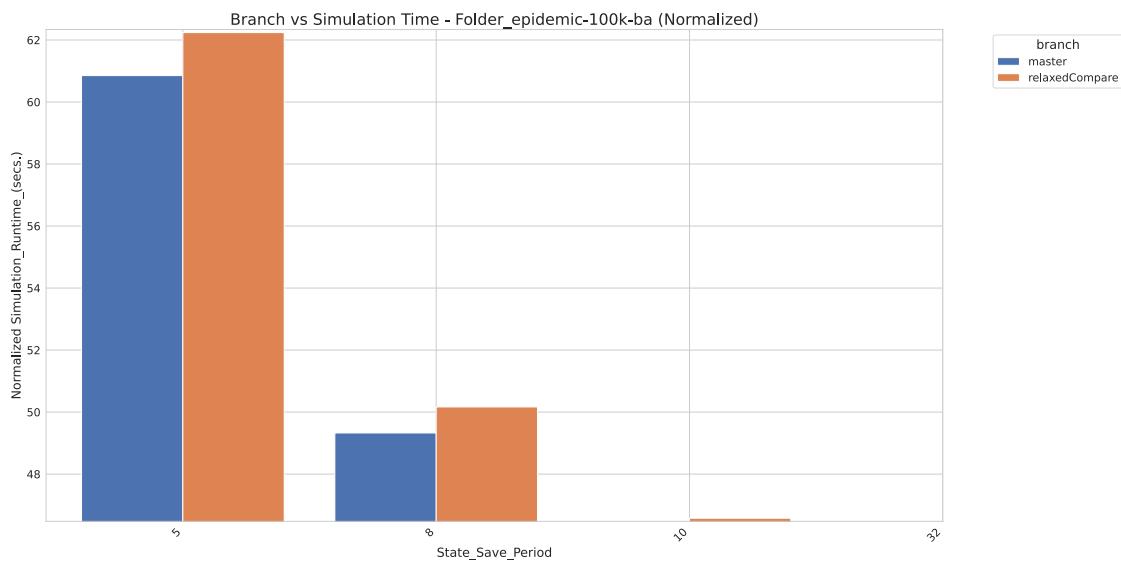


Figure 168: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

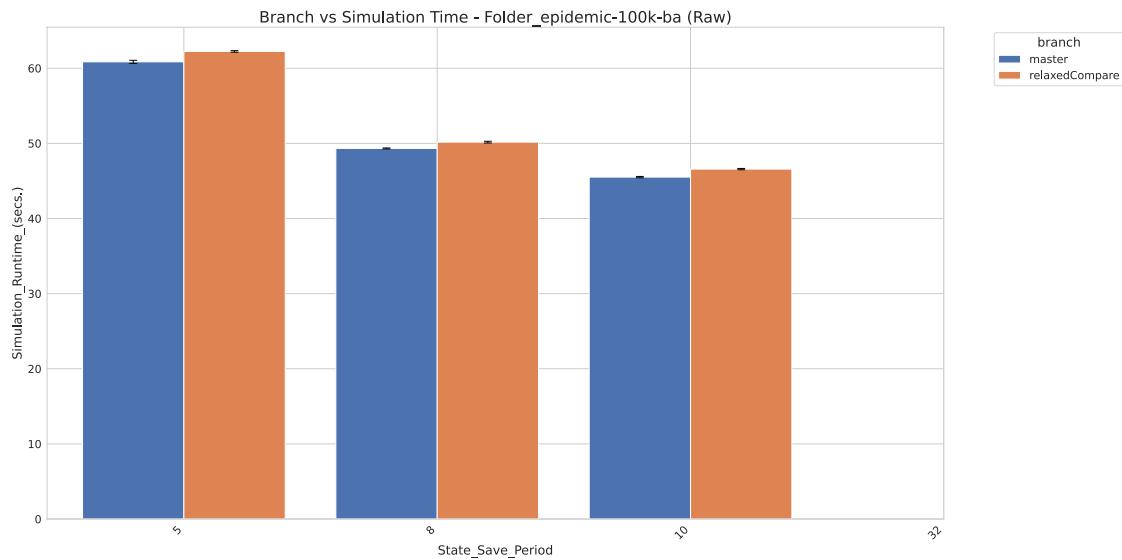


Figure 169: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

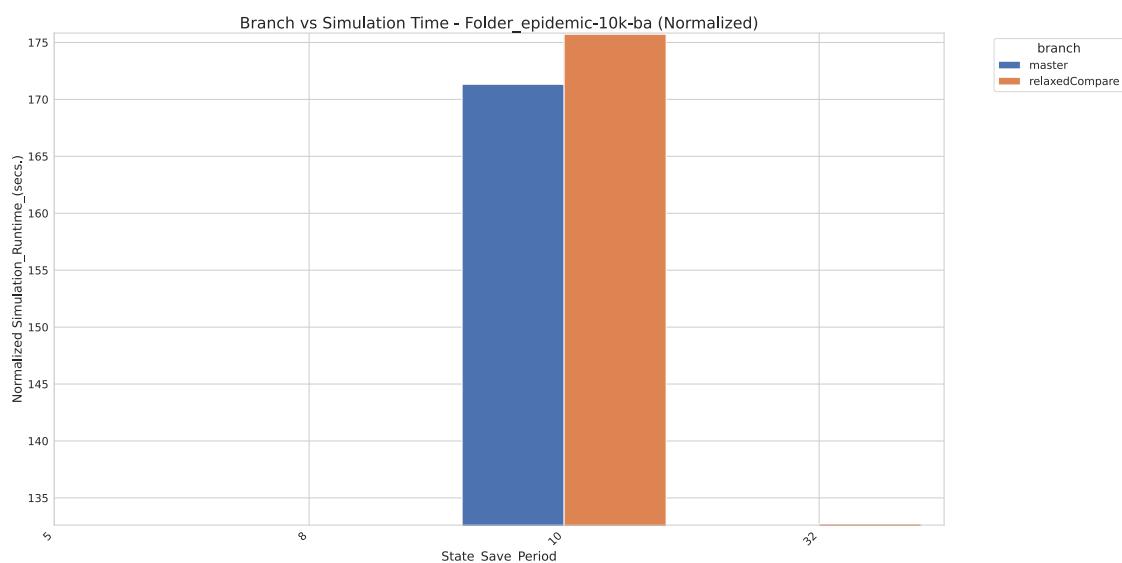


Figure 170: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

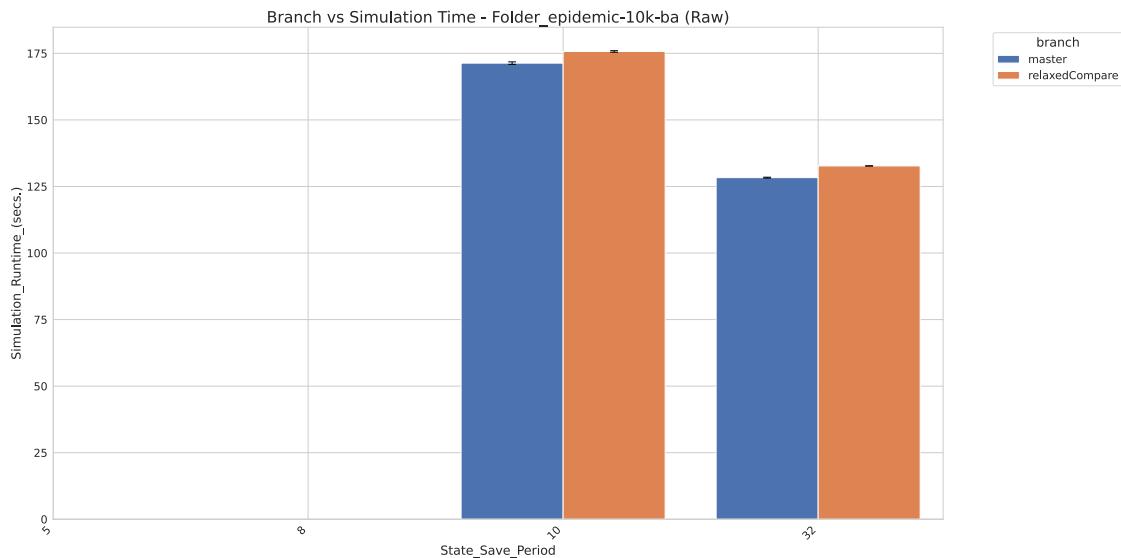


Figure 171: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

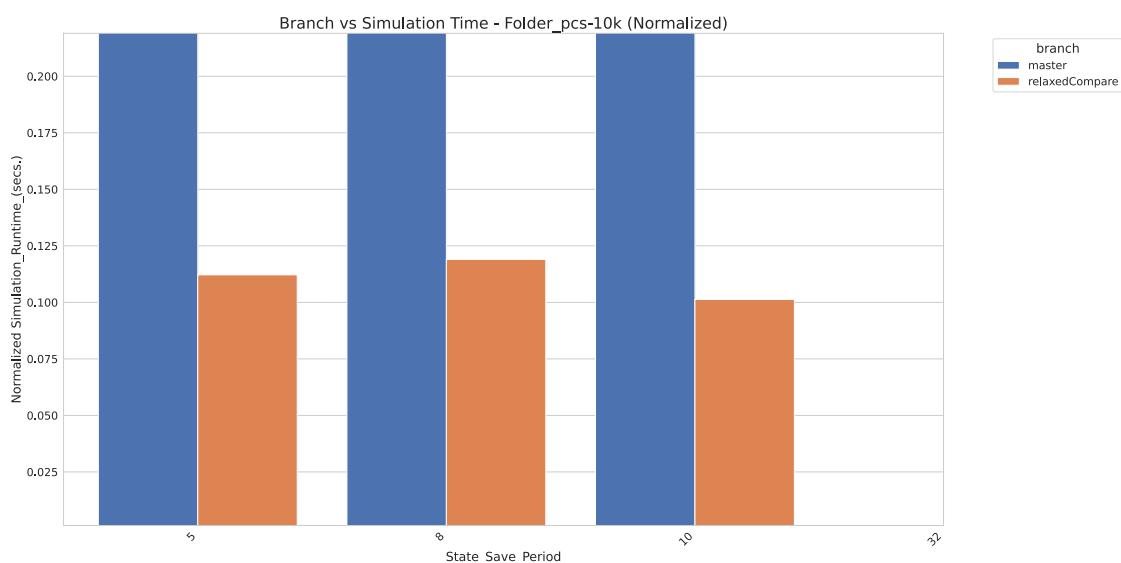


Figure 172: Branch vs Simulation Time_Folder_pcs-10k_normalized

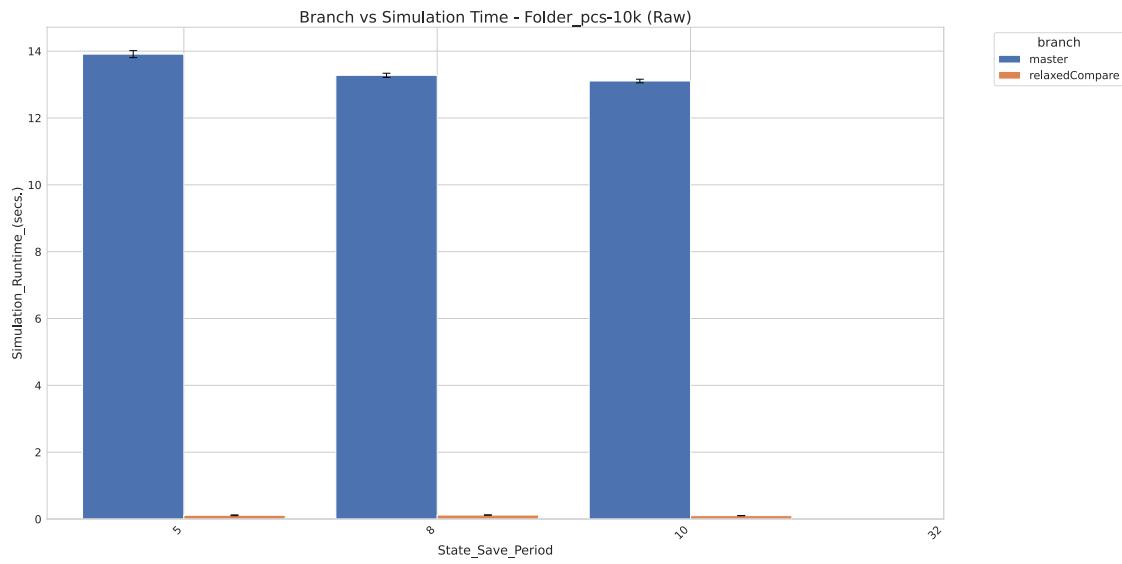


Figure 173: Branch vs Simulation Time_Folder_pcs-10k_raw

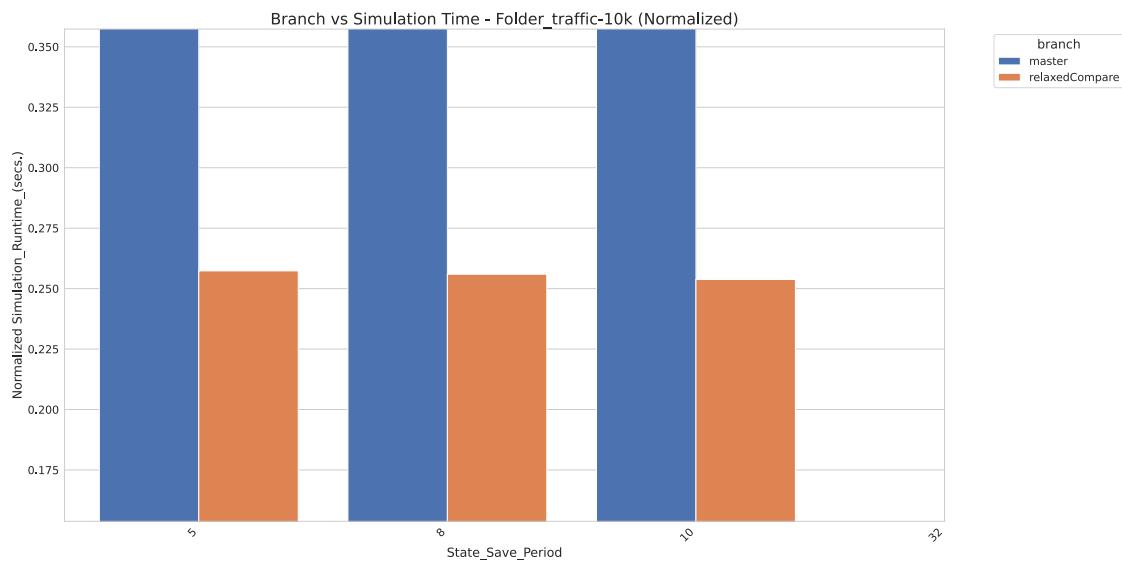


Figure 174: Branch vs Simulation Time_Folder_traffic-10k_normalized

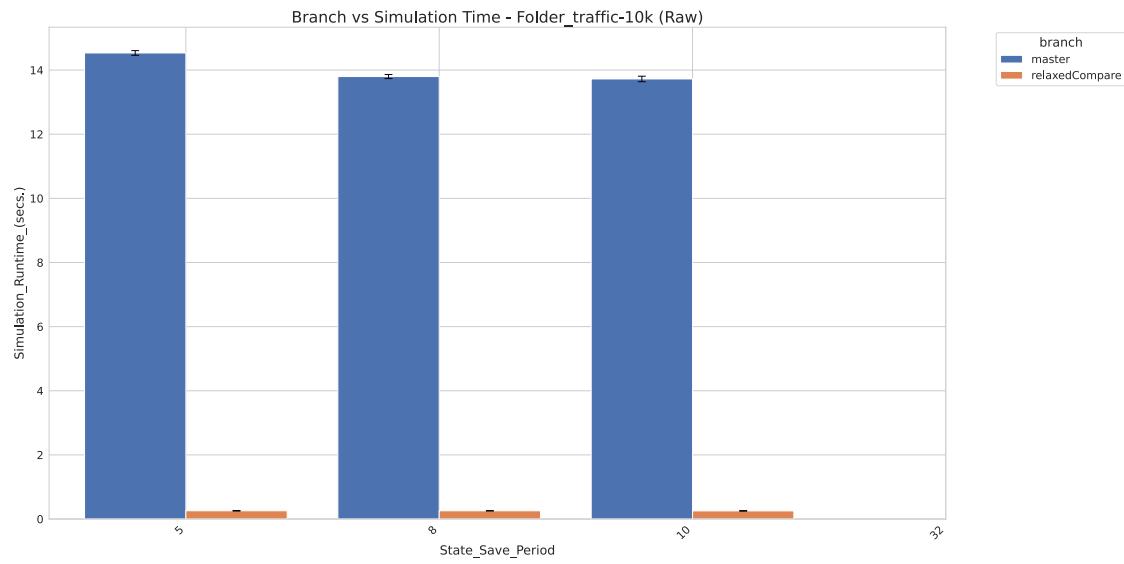


Figure 175: Branch vs Simulation Time_Folder_traffic-10k_raw

12 SIMD_local

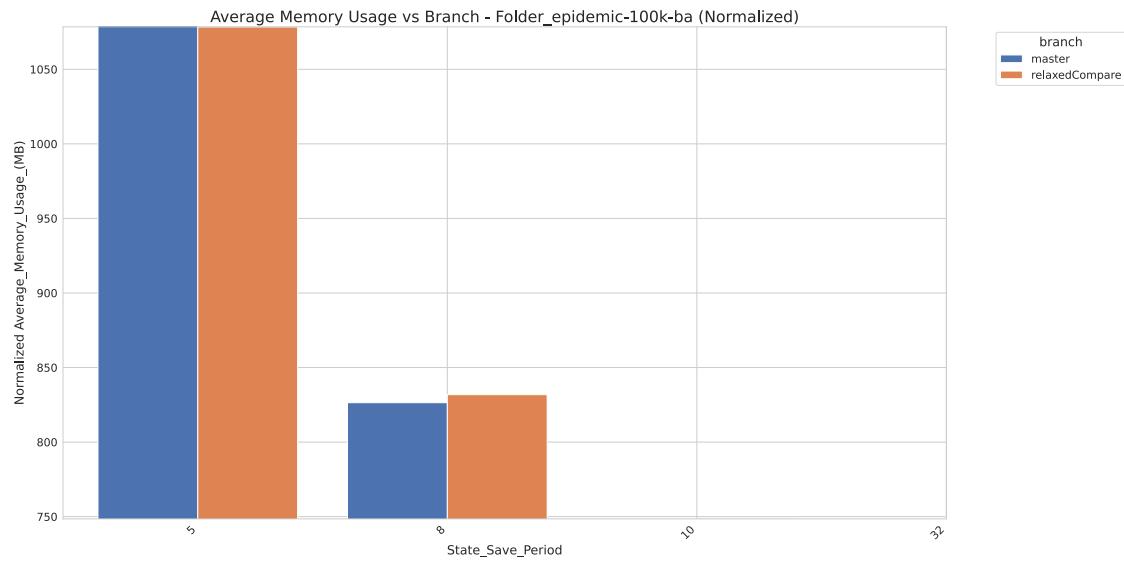


Figure 176: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_normalized

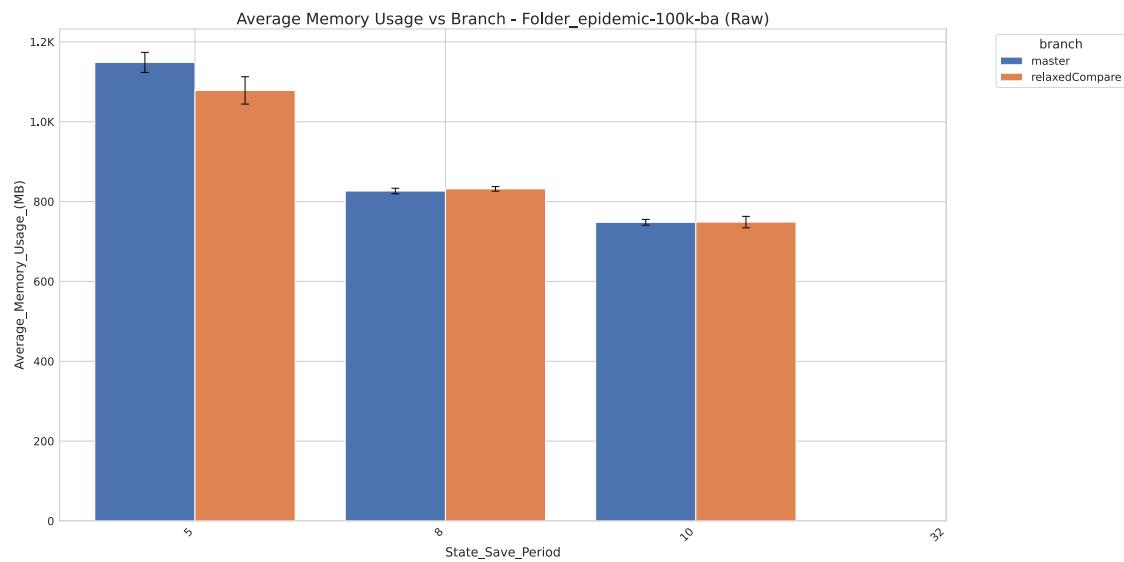


Figure 177: Average Memory Usage vs Branch_Folder_epidemic-100k-ba_raw

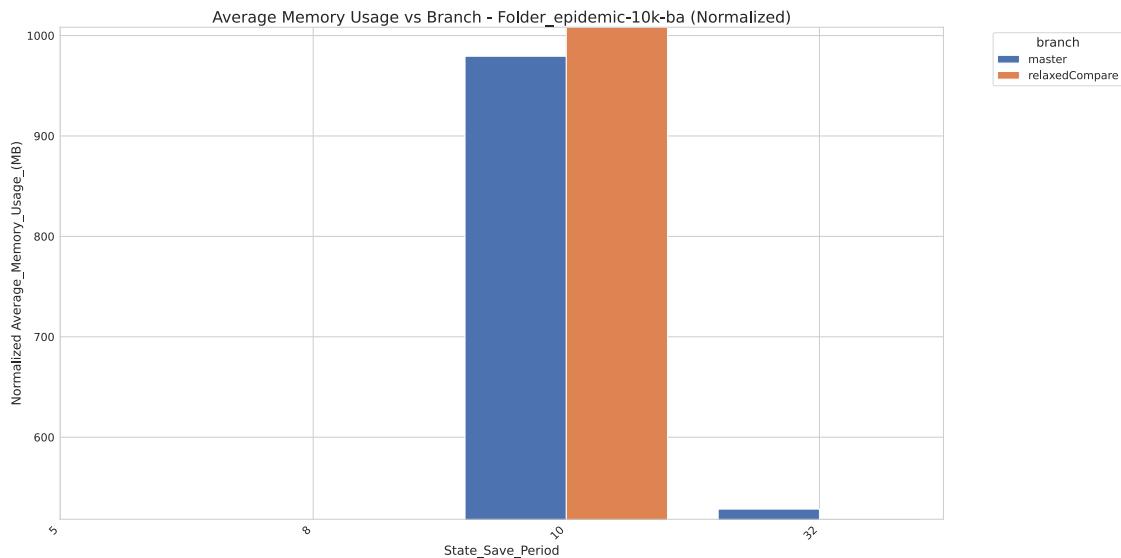


Figure 178: Average Memory Usage vs Branch_Folder.epidemic-10k-ba_normalized

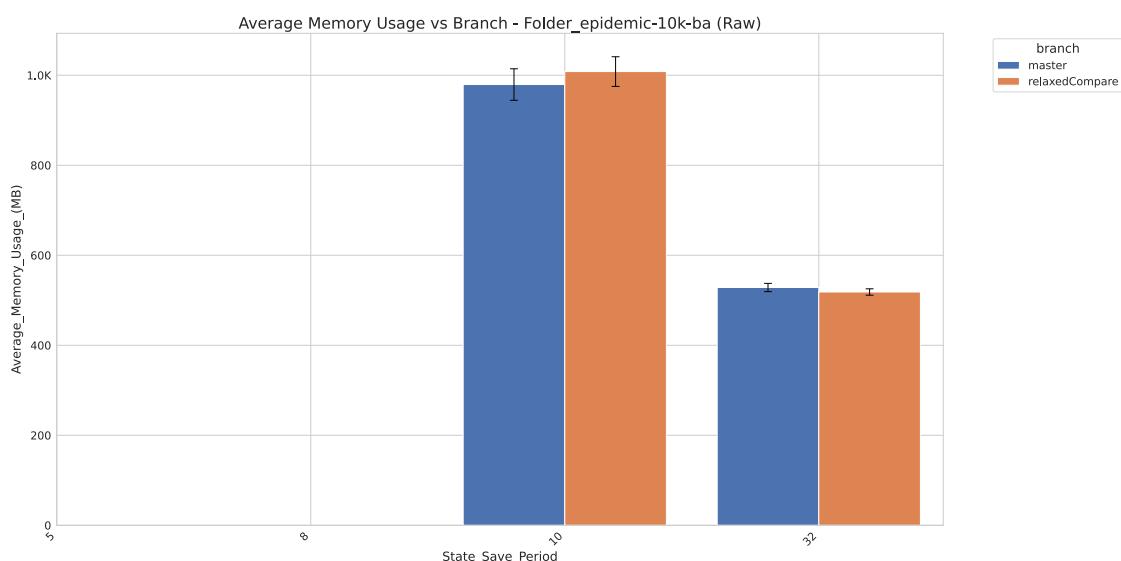


Figure 179: Average Memory Usage vs Branch_Folder.epidemic-10k-ba_raw

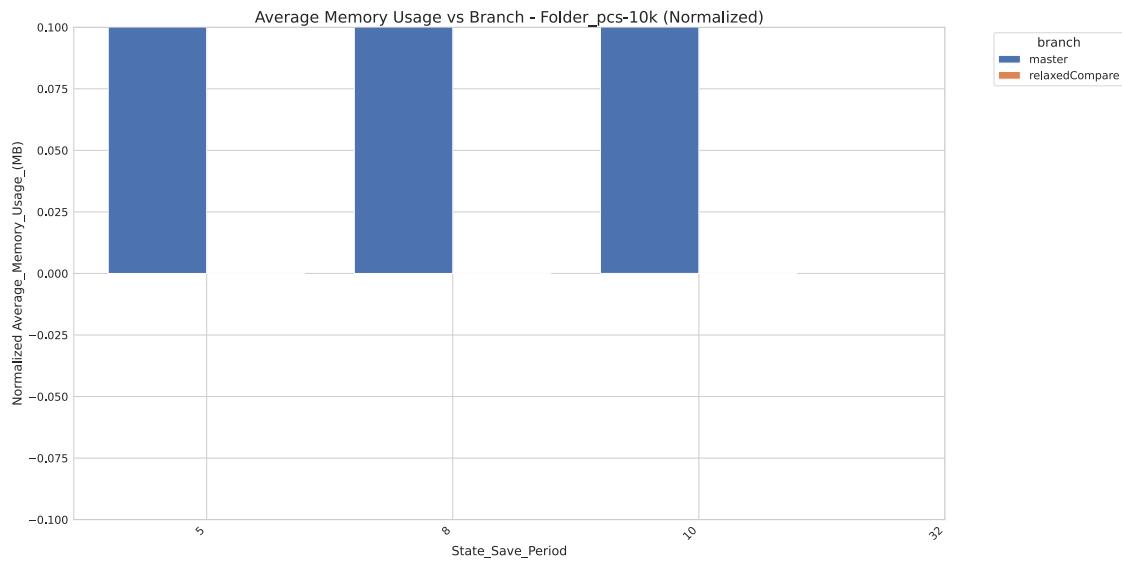


Figure 180: Average Memory Usage vs Branch_Folder_pcs-10k_normalized

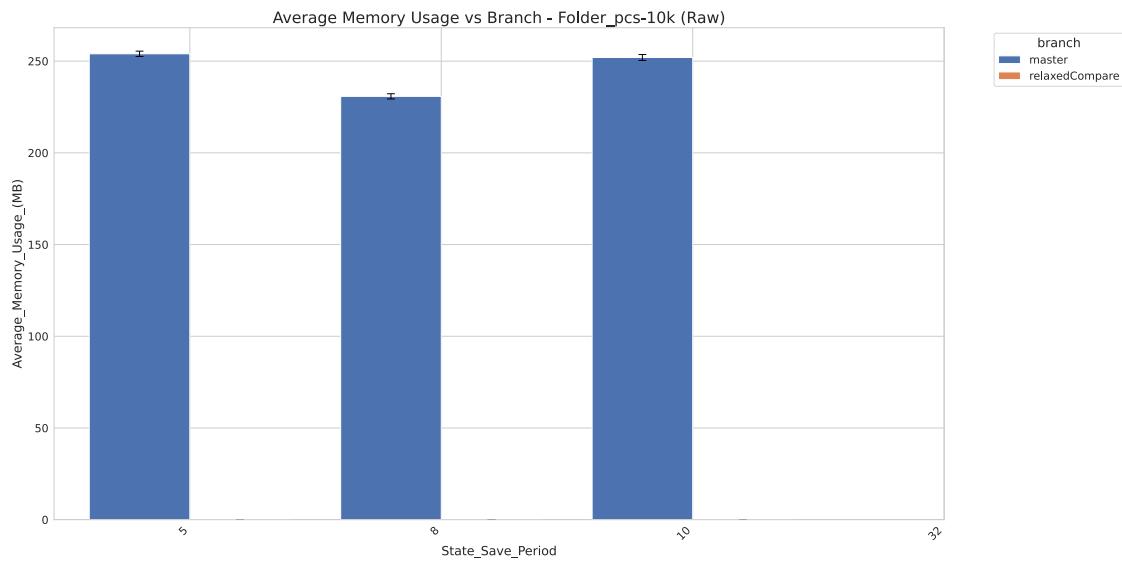


Figure 181: Average Memory Usage vs Branch_Folder_pcs-10k_raw

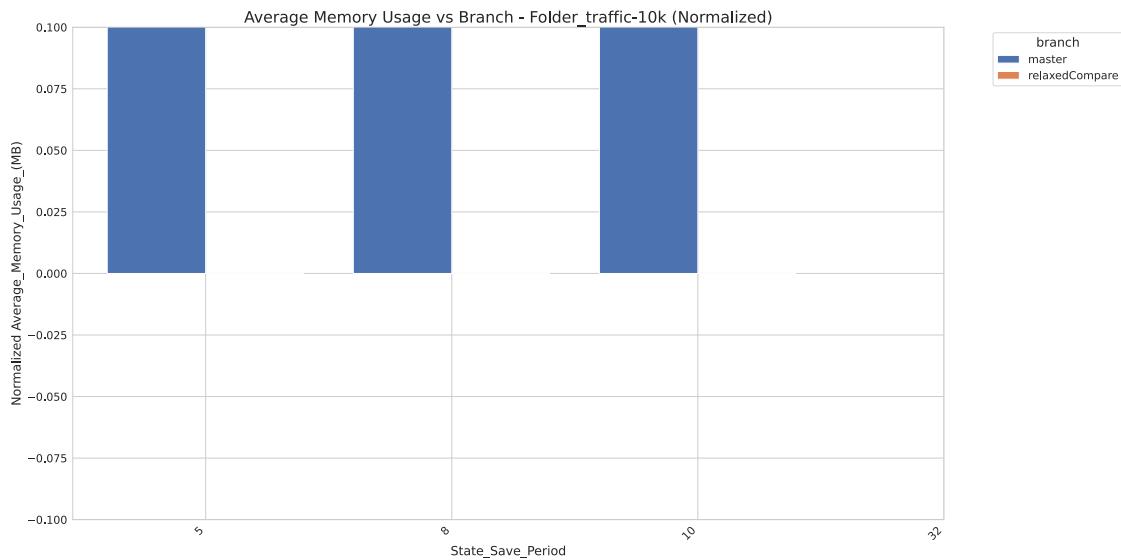


Figure 182: Average Memory Usage vs Branch_Folder_traffic-10k_normalized

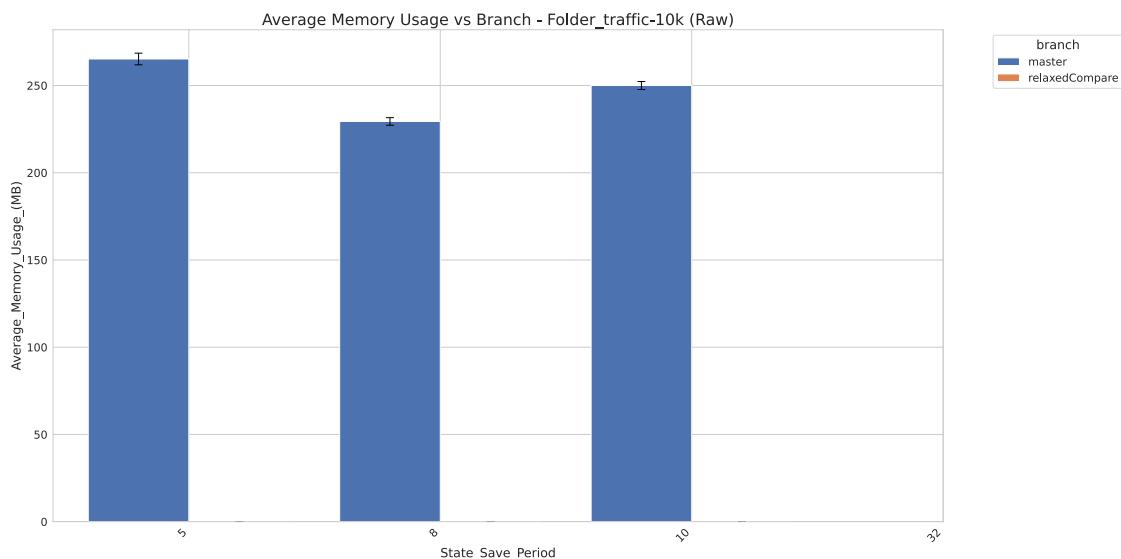


Figure 183: Average Memory Usage vs Branch_Folder_traffic-10k_raw

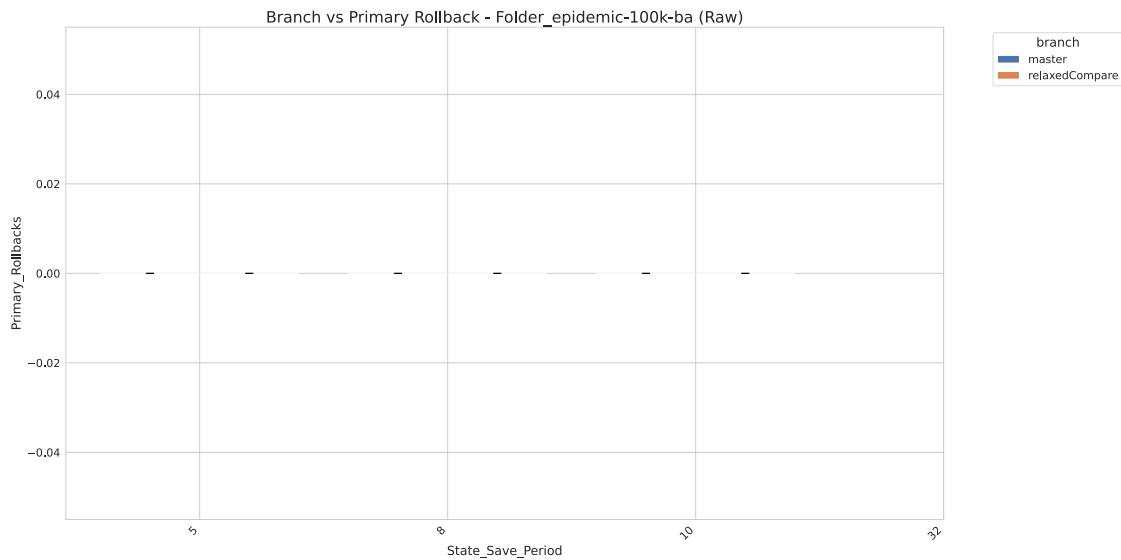


Figure 184: Branch vs Primary Rollback_Folder_epidemic-100k-ba_raw

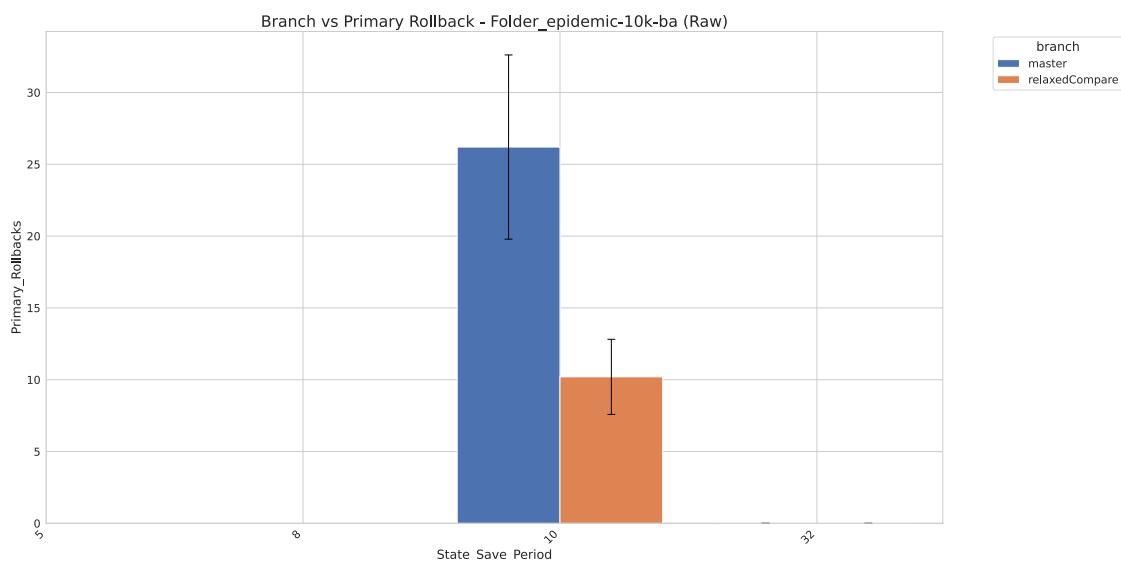


Figure 185: Branch vs Primary Rollback_Folder_epidemic-10k-ba_raw

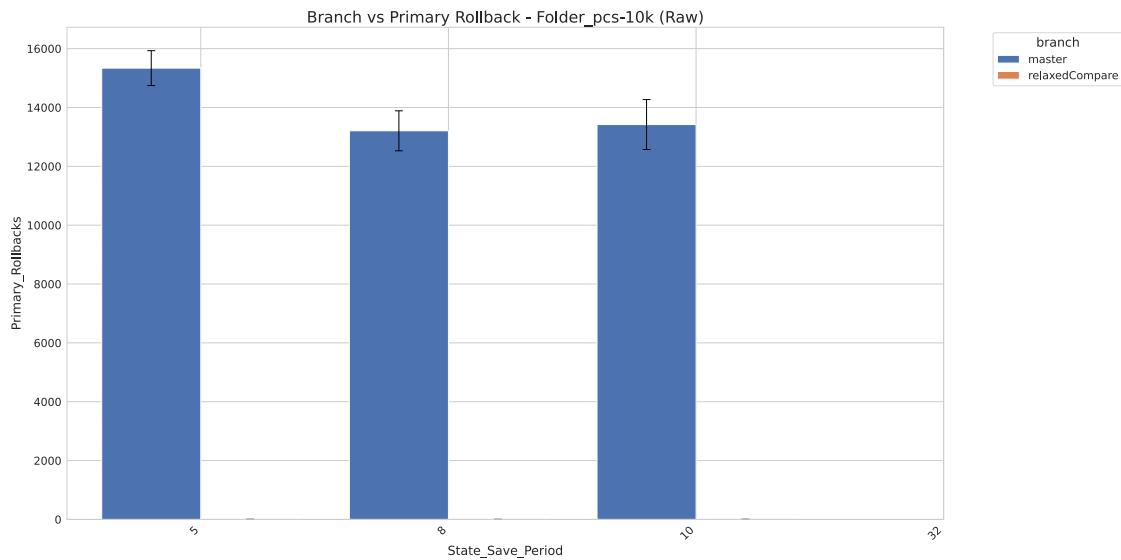


Figure 186: Branch vs Primary Rollback_Folder_pcs-10k_raw

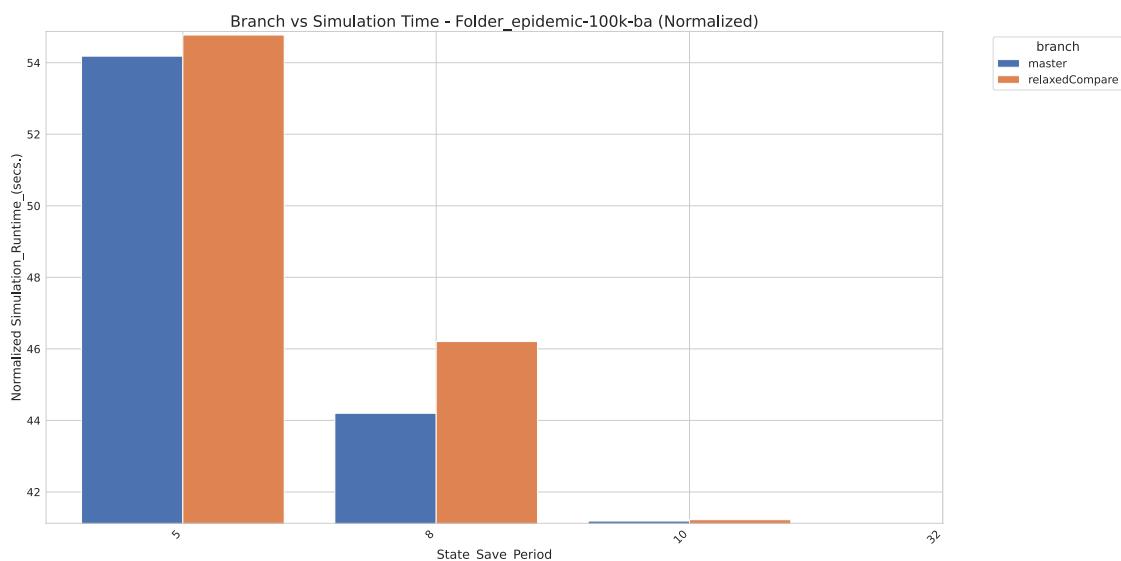


Figure 187: Branch vs Simulation Time_Folder_epidemic-100k-ba_normalized

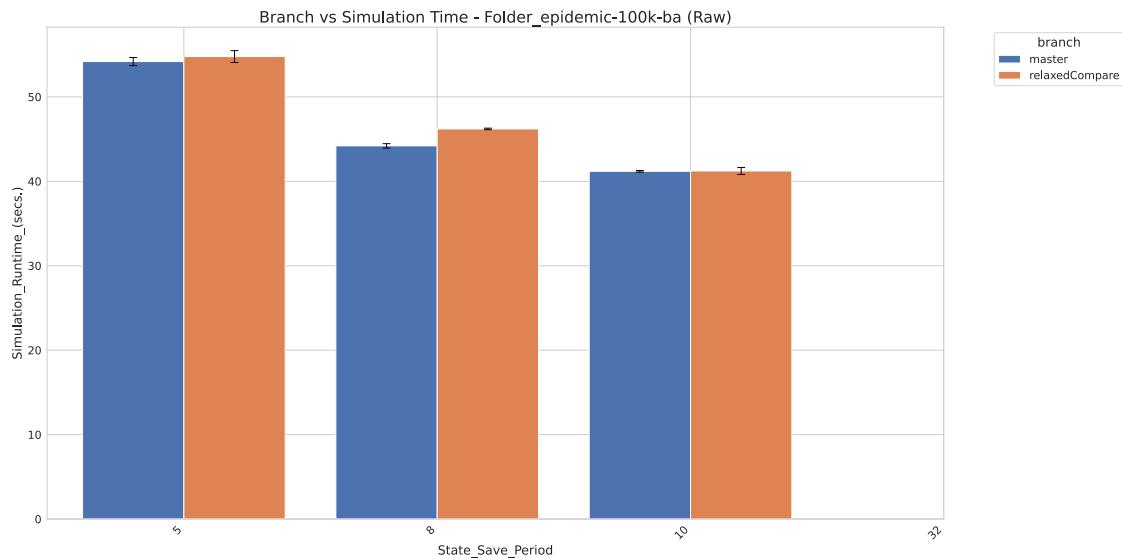


Figure 188: Branch vs Simulation Time_Folder_epidemic-100k-ba_raw

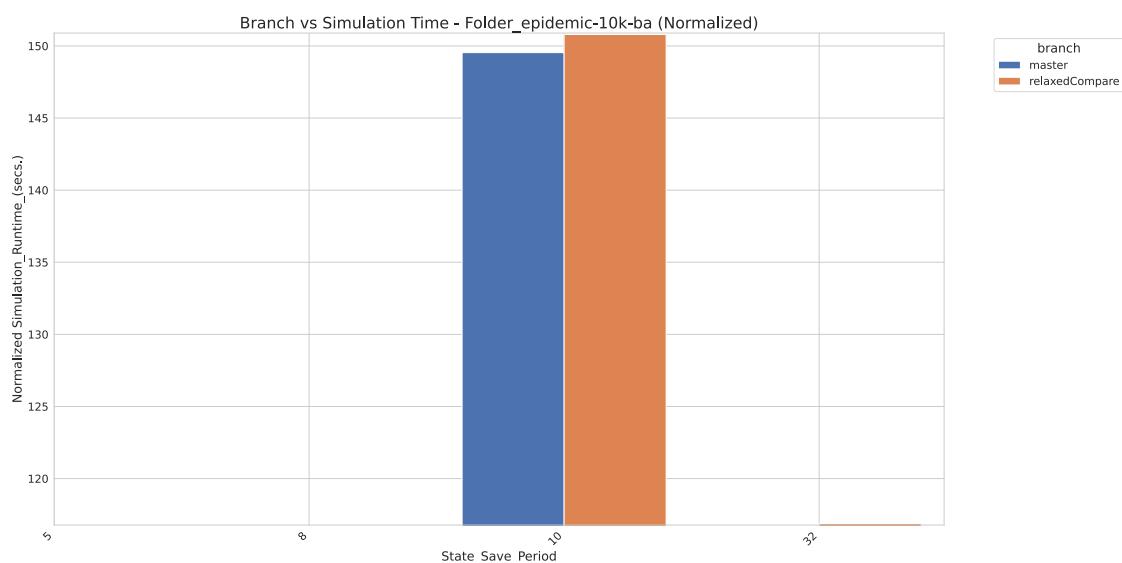


Figure 189: Branch vs Simulation Time_Folder_epidemic-10k-ba_normalized

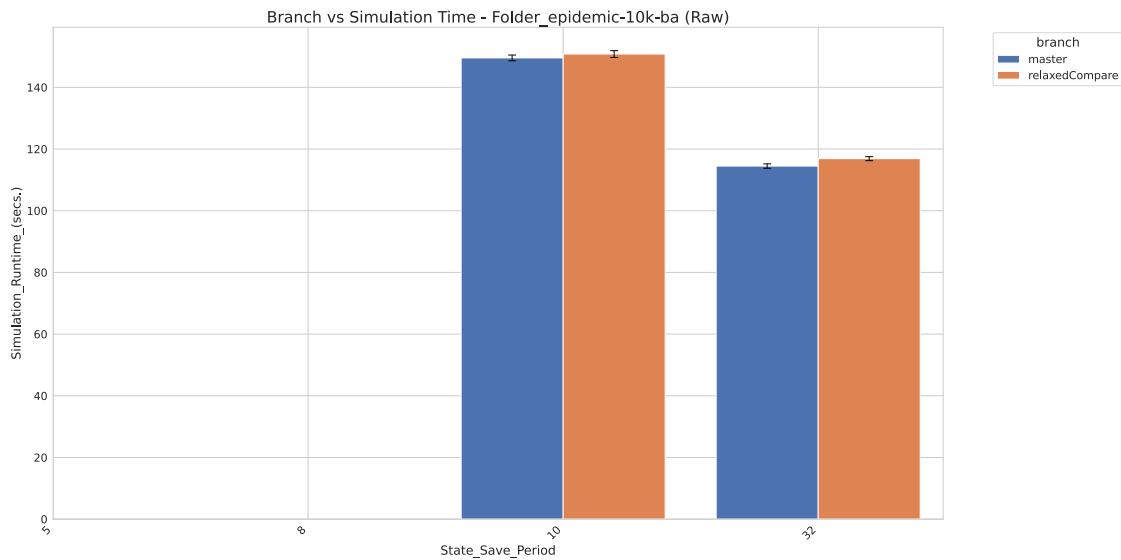


Figure 190: Branch vs Simulation Time_Folder_epidemic-10k-ba_raw

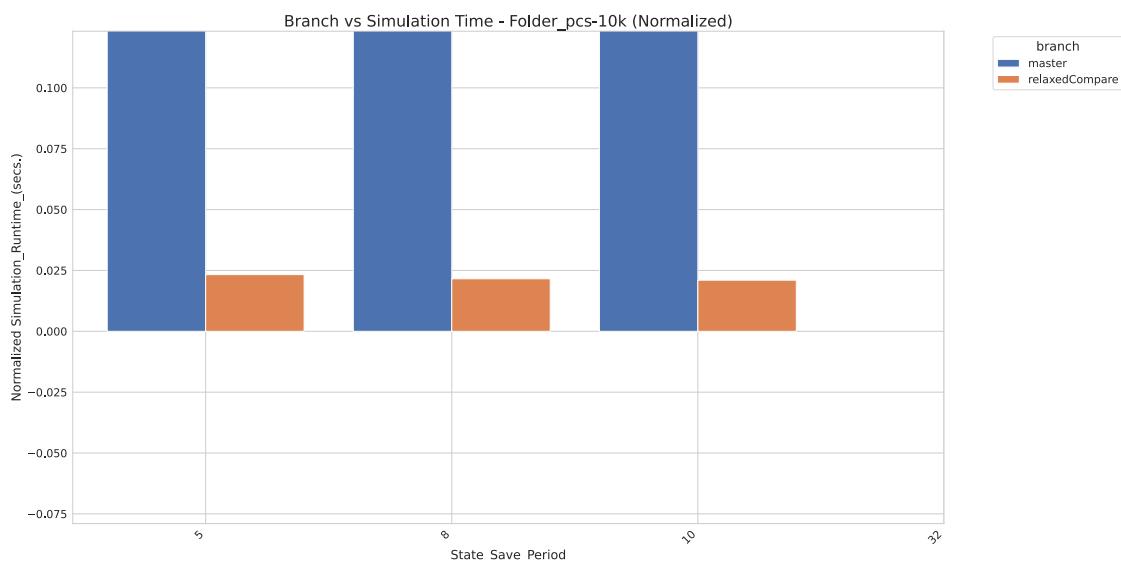


Figure 191: Branch vs Simulation Time_Folder_pcs-10k_normalized

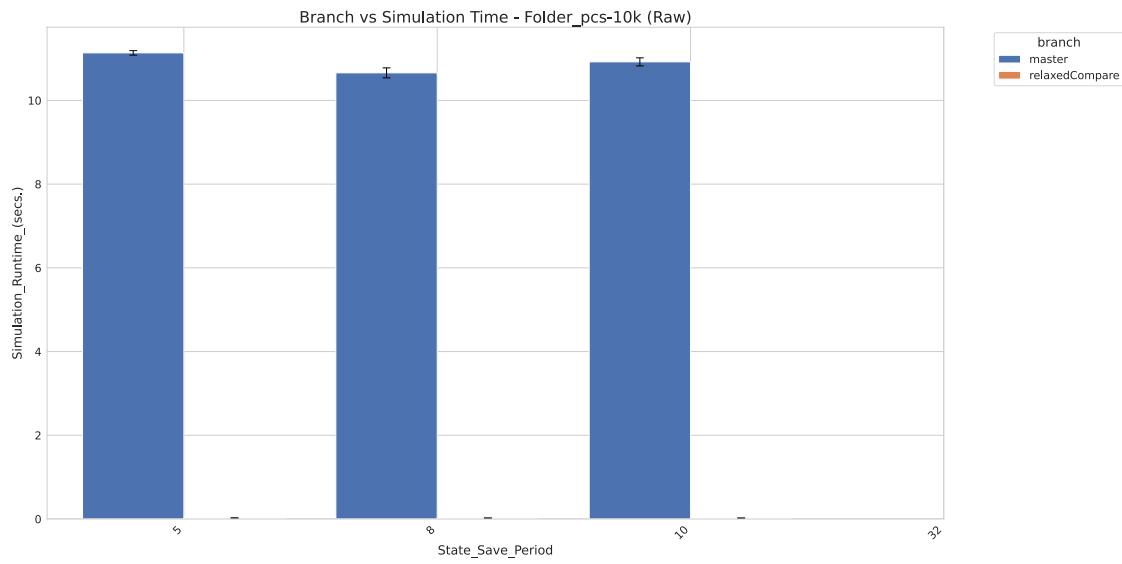


Figure 192: Branch vs Simulation Time_Folder_pcs-10k_raw

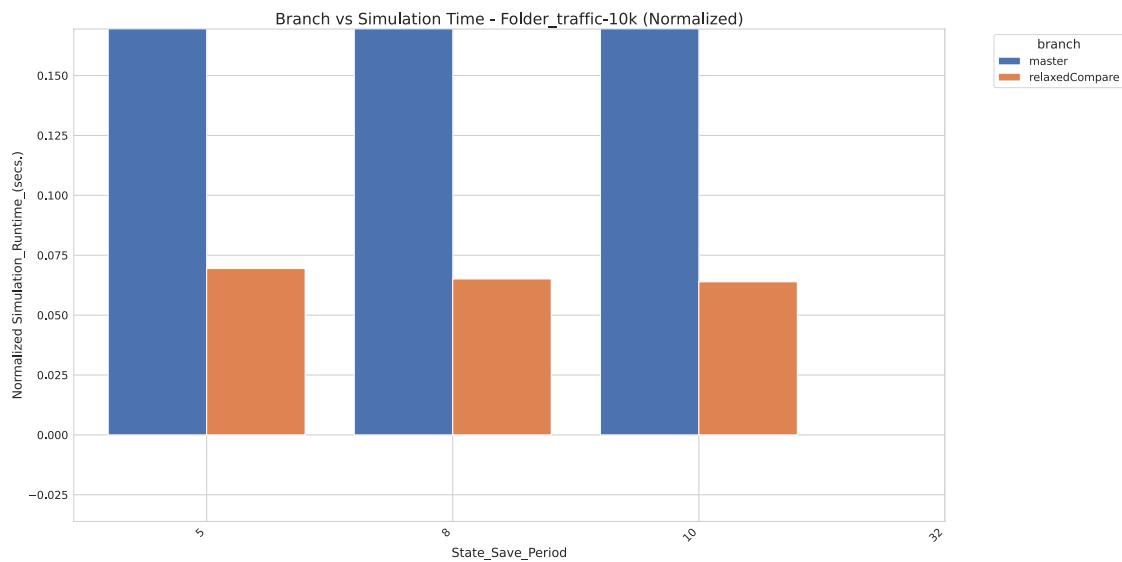


Figure 193: Branch vs Simulation Time_Folder_traffic-10k_normalized

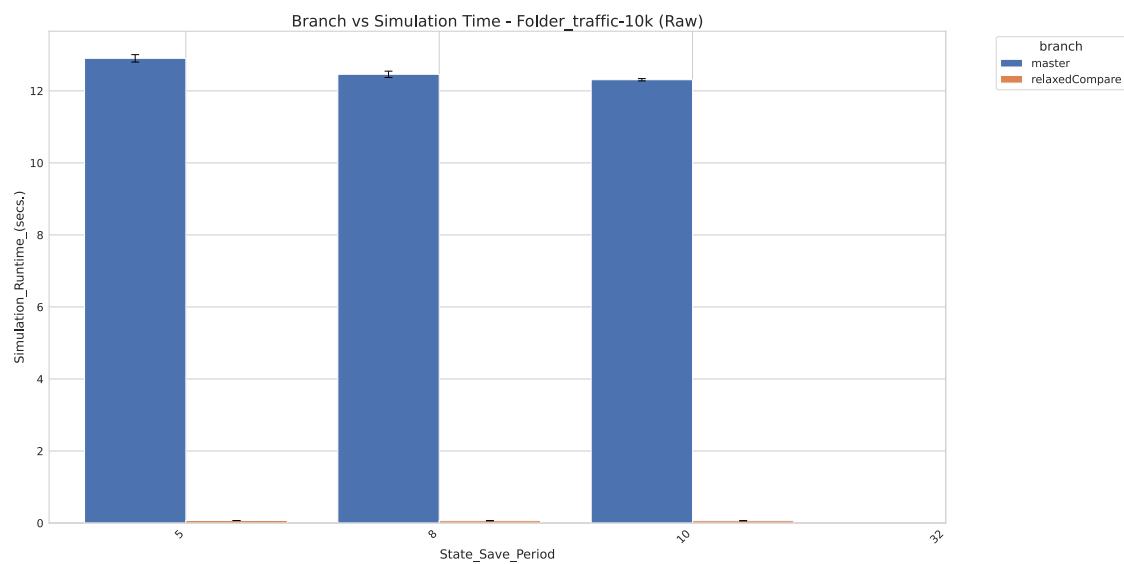


Figure 194: Branch vs Simulation Time_Folder_traffic-10k_raw

13 Top Performers

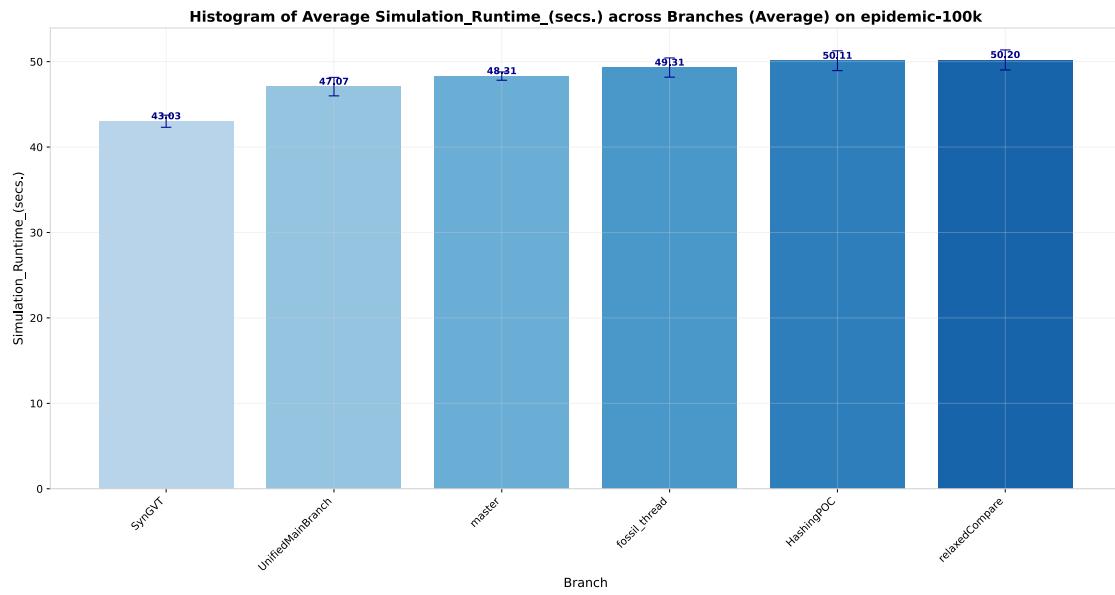


Figure 195: Epidemic100kPerformers_(Average)

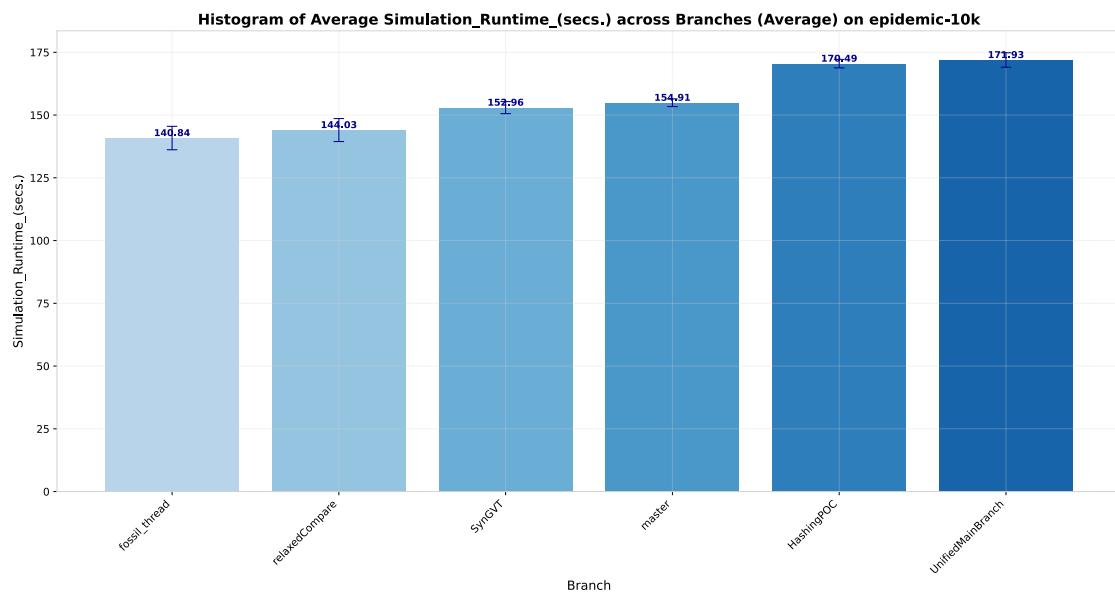


Figure 196: EpidemicPerformers_(Average)

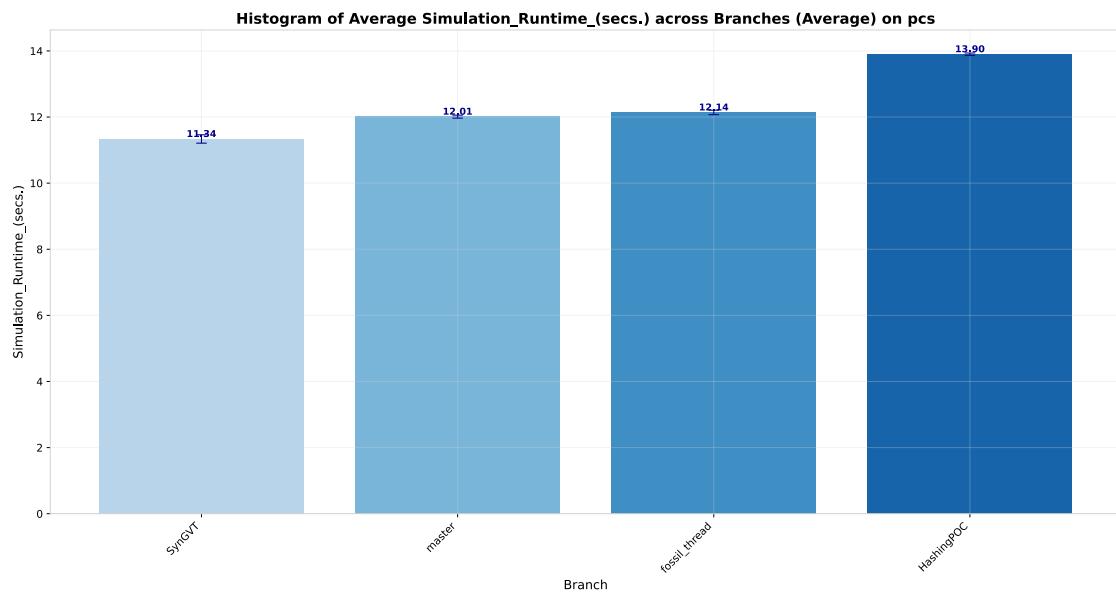


Figure 197: PCSPerformers_(Average)

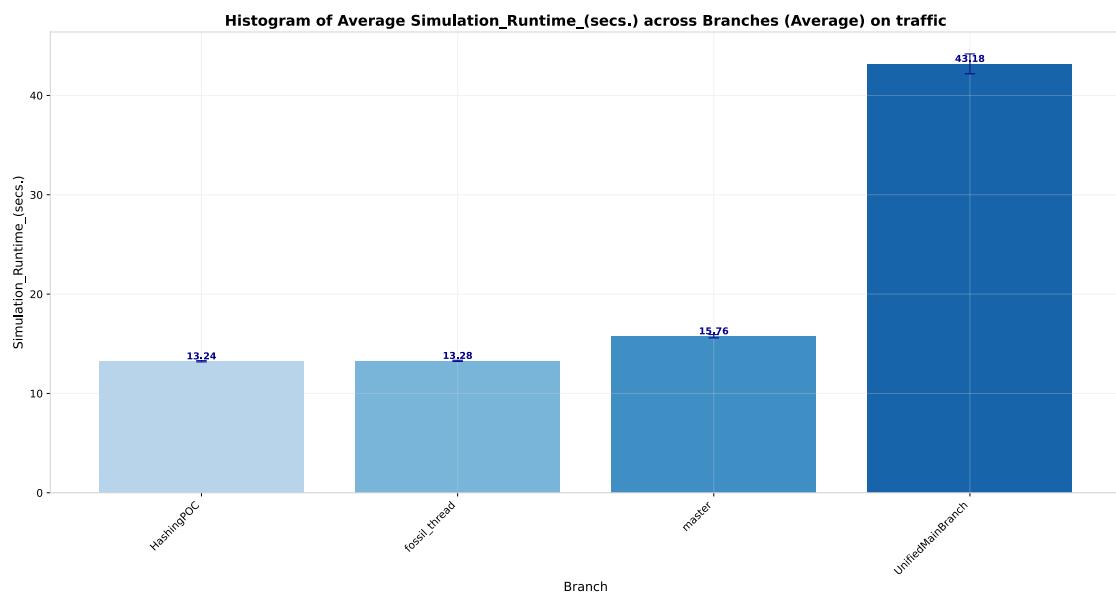


Figure 198: TrafficPerformers_(Average)

References

- [1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [2] Christopher L Barrett, Keith R Bisset, Stephen G Eubank, Xizhou Feng, and Madhav V Marathe. Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12, 2008.
- [3] Kalyan S Perumalla and Sudip K Seal. Discrete event modeling and massively parallel execution of epidemic outbreak phenomena. *Simulation*, 88(7):768–783, 2012.
- [4] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.