Comprehensive experimental design across all phases

Experiment	Configuration	Parameter Settings	Evaluation Purpose
P1_E1	Baseline Configuration	Standard configuration	Control condition for statistical comparisons against all experimental configurations.
P2_E1	parallel_4_lookahead_default	DefaultMaxParallelism = 4 DefaultMaxLookAhead = 24	Evaluate energy efficiency when reducing thread count to mitigate resource contention, as fewer active threads can result in energy savings when hardware resources are oversubscribed [1].
P2_E1	parallel_16_lookahead_default	DefaultMaxParallelism = 16 DefaultMaxLookAhead = 24	Assess energy consumption patterns when increasing parallelism beyond optimal levels, as the energy consumption of a processor can increase substantially with additional active cores [1].
P2_E1	parallel_default_lookahead_12	DefaultMaxParallelism = 8 DefaultMaxLookAhead = 12	Examine how reduced lookahead affects energy efficiency, as the granularity of work assignment significantly impacts thread control overhead and overall performance [2].
P2_E1	parallel_default_lookahead_48	DefaultMaxParallelism = 8 DefaultMaxLookAhead = 48	Evaluate energy implications of increased lookahead to ensure adequate work chunks for balanced thread utilization while minimizing scheduling overhead [2].
P2_E2	reduced	enableTracing = true consoleLogging = false logLevel = Warning	Evaluate the energy and performance impact of reduced log verbosity while keeping trace instrumentation enabled, since logging activity is known to introduce CPU, memory, and I/O overhead [3].
P2_E2	minimal	enableTracing = false consoleLogging = false logLevel = Error	Evaluate the energy and performance impact of disabling trace instrumentation and retaining only conservative logging, as prior evidence in mobile systems suggests that low logging rates are unlikely to affect energy consumption [4].
P2_E3	extended_expiration	maxCachedItems = 10000 expirationInSeconds = 300	Examine how extending cache duration affects performance and efficiency, supported by evidence that longer retention reduces request load and improves responsiveness in cached systems [5].
P2_E3	larger_cache	maxCachedItems = 20000 expirationInSeconds = 60	Assess how increasing cache size improves runtime efficiency, based on findings that cache size has the strongest influence on energy and performance among cache configuration parameters [6].
P2_E4	no_compression	enableResponseCompression = false	Test disabling compression, as [7] shows compression can take more time and energy than transferring uncompressed data.
P2_E5	MemoryConservation	gcServer = 1 gcConcurrent = 1 GCConserveMemory = 7 GCLOHThreshold = 140000	Test whether increasing GC frequency and compaction reduces memory usage under constrained conditions, as similar tuning has impacted memory efficiency in embedded systems [8].
P2_E5	DynamicAdaptation	gcServer = 1 gcConcurrent = 1 GCDynamicAdaptationMode = 1 GCRetainVM = 0	Evaluate whether enabling runtime heap resizing improves responsiveness under memory pressure, following similar adaptive strategies shown effective in Java VMs [9].
P2_E5	ThreadEfficient	gcServer = 1 gcConcurrent = 1 GCNoAffinitize = 1 GCHeapCount = 4	Test whether tuning GC thread count improves throughput, as poor GC thread sizing can cause performance loss [10].
P2_E5	ReducedFootprint	gcServer = 1 gcConcurrent = 1 GCRetainVM = 0 GCHighMemPercent = 46 GCConserveMemory = 5	Assess whether limiting retained memory and heap growth reduces footprint over time, as compaction-based strategies have helped reduce memory fragmentation [8].
P3_E1	Carbon-Optimized	Combined optimal settings from Phase 2	Evaluate combined effectiveness of the best-performing configuration from each Phase 2 category, provided they achieve at least 5% SCI score reduction.

References

[1] Stephen L. Olivier, Allan K. Porterfield, Sridutt Bhalachandra, and Jan F. Prins. Power measurement and concurrency throttling for energy reduction in OpenMP programs. In *Proceedings of the 2013 IEEE 27th International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, IPDPSW '13, pages 884–891, Boston, MA, USA, 2013. IEEE.

[2] Performance Optimisation and Productivity Centre of Excellence. Chunk/task grain-size trade-off (parallelism/overhead), 2025.

[3] Rui Ding, Hucheng Zhou, Jian-Guang Lou, Hongyu Zhang, Qingwei Lin, Qiang Fu, Dongmei Zhang, and Tao Xie. Log2: A cost-aware logging mechanism for performance diagnosis. In *Proceedings of the 2015 USENIX Annual Technical Conference*, USENIX ATC '15, pages 139–150, Santa Clara, CA, USA, 2015. USENIX Association.

[4] Shaiful Chowdhury, Silvia Di Nardo, Abram Hindle, and Zhen Ming (Jack) Jiang. An exploratory study on assessing the energy impact of logging on Android applications. *Empirical Software Engineering*, 23(3):1422–1461, 2018.

[5] Joon Jung, Daniel Cicalese, Amogh Dhamdhere, Matthew Luckie, Jay Kroll, David Clark, and kc Claffy. Cache me if you can: Effects of DNS time-to-live. In *Proceedings of the 2019 Internet Measurement Conference*, IMC '19, pages 313–326, Amsterdam, Netherlands, 2019. ACM.

[6] Yuan Cai, Marcus T. Schmitz, Alireza Ejlali, Bashir M. Al-Hashimi, and Sudhakar M. Reddy. Cache size selection for performance, energy and reliability of time-constrained systems. In *Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '05, pages 923–926. IEEE, 2005.

[7] Aleksandar Milenković, Armen Dzhagaryan, and Martin Burtscher. Performance and energy consumption of lossless compression/decompression utilities on mobile computing platforms. In *Proceedings of the 2013 IEEE 21st International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, MASCOTS '13, pages 254–263, San Francisco, CA, USA, 2013. IEEE.

[8] G. Chen, R. Shetty, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and M. Wolczko. Tuning garbage collection in an embedded Java environment. In *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture*, HPCA '02. IEEE, 2002.

[9] Fan Yang, Dhruva Gupta, Prashant Soman, Indranil Gupta, and Chandra Krintz. Automatic heap sizing: Taking real memory into account. In *Proceedings of the 2001 International Symposium on High-Performance Computer Architecture*, HPCA '01, pages 155–166. IEEE, 2001.

[10] Ram Lakshmanan. Java performance tuning: Adjusting GC threads for optimal results, 2024.