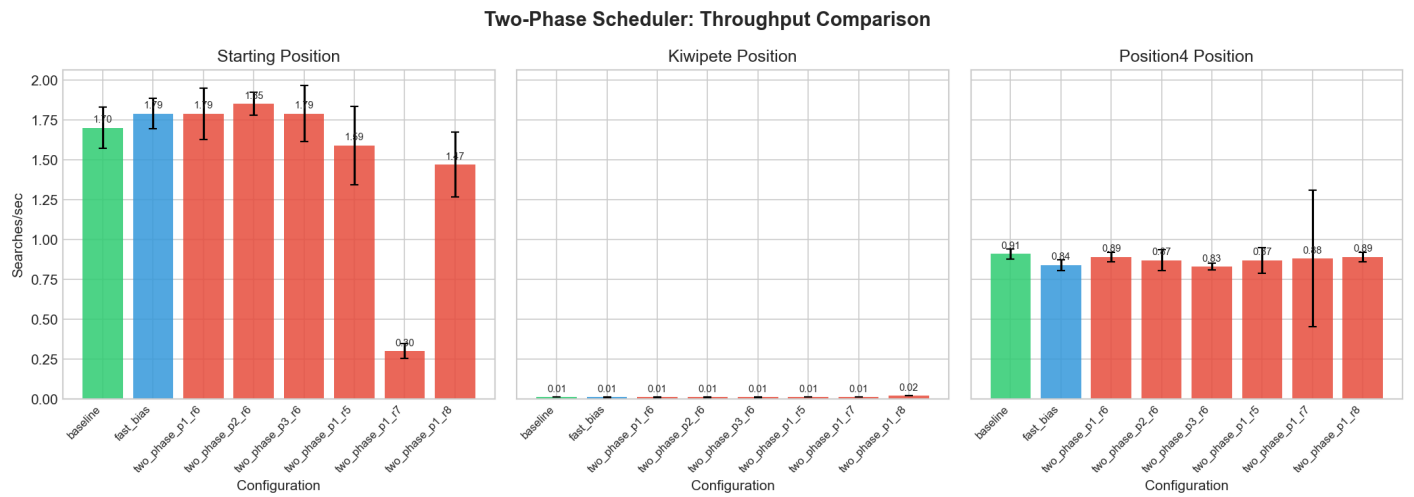


Two-Phase Scheduler Summary (v0.5.0)

Motivation: My v0.4.0 QoS experiments showed a **12.8x P/E gap** on M1 Pro, with mixed scheduling achieving only 65% expected throughput. Rather than fight the OS scheduler, I built a work-aware dispatch layer.

Approach: Probe root moves to estimate subtree size, run heavy subtrees on P-cores first, then hand E-cores the tightened alpha bound so they prune aggressively.

Results



Two-phase (red) vs baseline P-core only (green). Kiwipete's 48-move complexity shows the clearest win.

Position	Moves	Speedup	Config
Kiwipete	48	1.55x	probe=1, ratio=0.8
Starting	20	1.09x	probe=2, ratio=0.6
Position4	6	1.00x	skip two-phase

Failure discovered: ratio=0.7 on starting position -> 0.18x (5.6x slower). E-cores became bottleneck

Recommendation

- **30+ moves:** use two-phase with ratio 0.8
- **10-30 moves:** Use ratio 0.6
- **<10 moves:** Skip two-phase entirely

Key Insight

The alpha-bound handoff is what makes this work. Without it, E-cores search blind and become the critical path.

For methodology, failure analysis, and reproduction steps, see [two_phase_detailed.md](#).