Meeting Note 4

Date: 2026-01-02 **Time:** 14:00 - 15:00 **Location:** Supervisor's Office (PQ706) **Attendees:** LIN Zhanzhi, CAO Yixin (Supervisor)

1. Objectives

- Provide a comprehensive progress report on the completed sequential and parallel modules.
- Present the benchmarking results showing performance superiority over `std::sort`.
- Validate the scope and structure for the upcoming Interim Report.
- Discuss technical decisions regarding SIMD and parallel architecture.

2. Progress Report

- **Algorithmic Core**:
  - Successfully implemented **3-way partitioning** with "Median-of-5" pivot selection, achieving ~20% fewer swaps.
  - Integrated **Adaptive Run Merging** (TimSort-style) to handle pre-sorted data efficiently.
  - Implemented **Introsort Strategy** (fallback to HeapSort) to guarantee $O(N \log N)$ worst-case safety.
  - Added **Counting Sort** optimization for small integer types (`byte`, `short`).
- **Parallel Architecture**:
  - Deployed a **Work-Stealing Thread Pool** for dynamic load balancing.
  - Completed contention analysis and mutex optimization.
- **Robustness**:
  - Ensured namespace safety and fixed integer overflow issues for large arrays.
  - Verified generic support with custom comparators.
- **Infrastructure**:
  - Established a custom **Benchmarking Harness** with "Destructive Testing" support.
  - Performed rigorous sample size analysis to ensure statistical significance.

3. Discussion & Feedback

Key Discussion Points

- **Sequential Performance**: Reviewed the data showing DPQS outperforming `std::sort` on random integers and achieving massive speedups on structured data (e.g., Organ Pipe). Supervisor acknowledged the efficacy of the "Run Merging" strategy.
- **Parallel Scaling**: Discussed the "Strong Scaling" results. The linear scaling up to 4 threads and saturation beyond 8 threads (due to Memory Bandwidth) was accepted as a valid finding for the Interim Report.
- **SIMD Feasibility**: Presented the decision to prioritize Parallelism over SIMD for Semester 1. Supervisor agreed that SIMD complexity is high and fitting as a "stretch goal" for Semester 2.
- **Comparison Targets**: Confirmed that comparing against `std::sort` and `std::stable_sort` is sufficient for the interim phase. Future work can include comparisons with TBB or OpenMP.

Supervisor Feedback

> "The algorithmic tuning, especially the Adaptive Run Merging, is a strong point. Make sure to highlight *why* 3-way partitioning helps with duplicates in your report."

> "Your identification of the 'Memory Wall' in the parallel section is critical. Ensure your Interim Report explains this hardware bottleneck clearly—don't just show the graph, explain the saturation."

> "The scope for the Interim Report is appropriate. The focus on 'Implementation & Validation' is solid. You don't need AVX-512 yet; getting the thread pool working correctly was the right priority."

4. Issues & Challenges

- **Parallel Overhead**: Discussed the challenge of managing thread overhead for small-to-medium arrays.
  - *Resolution:* Masked by the dynamic threshold ($N < 4096$ falls back to sequential), which effectively amortizes the cost.
- **SIMD Complexity**: Acknowledged that AVX-512 implementation is complex.
  - *Resolution:* Formally moved to Semester 2 plan as "Advanced Optimization".

5. Action Items

- □ Finalize the Interim Report based on the approved outline. (Due: 2026-01-09)
- □ Prepare the Interim Presentation slides, focusing on the "Speedup Analysis" graphs. (Due: 2026-01-09)
- □ Refine the work-stealing deque implementation for better load balancing in edge cases. (Due: Feb 2026)
- □ Begin preliminary investigation into AVX-512 intrinsics for the "Future Work" section. (Due: Feb 2026)

6. Next Meeting Plan

- **Target Date:** 2026-02-27 (Semester 2 Start)
- **Focus:** Detailed plan for SIMD implementation and advanced parallel optimizations.