

Parallel Sort Verifier

主題意義：那我們為什麼要做這件事？

在大型資料管線裡，排序結果往往被拆開、重排、再合併。
只要 **一筆** 元素掉到錯誤位置，後續演算法就可能整批失效。
我想寫一個「驗證器」， **不用重新排序**，就能：

- 1. 判斷結果是否全域遞增
- 2. 若有問題，回報第一個錯位索引與錯位區段
- 3. 10 億筆資料在 1 分鐘內完成檢查

核心數學條件

排序正確 \Leftrightarrow 對所有索引 i

$$a_i \leq a_{i+1}$$

若某段 $[l, r]$ 出現

$$a_k > a_{k+1}, \quad l \leq k < r$$

那你會看到，此段必為**錯位區段**。

所以，我會使用 **平行二分搜尋** 將 $[l, r]$ 逐步縮小到長度 $< K$ （預設 $K = 1024$ ），方便後續人工或自動修復。

三級驗證流程

Level	內容	時間複雜度
0	Local Scan ：各 Rank 掃自己區塊	$O(N/P)$
1	Boundary Check ：交換臨界值，檢查跨塊遞增	$O(P)$
2	Parallel Binary Locate ：對錯位塊做平行二分	$O(E \log \frac{N}{PK})$

$$T_{\text{total}} \approx \frac{N}{PB} + 2\alpha(P-1) + E \log_2\left(\frac{N}{PK}\right)$$

B ：每核記憶體頻寬； α ：訊息延遲； E ：錯位區段數。

平行策略一覽

技術	實作重點
MPI	<i>Master-Worker</i> 派送區塊； <code>MPI_Reduce</code> 聚合 <code>hasError</code> ；跨 Rank 交換 <code>max / min</code>
OpenMP	區塊內 <code>#pragma omp parallel for</code> 同步檢查 $a_i > a_{i+1}$
動態排程	若某 Rank 找到錯位 \rightarrow 立刻切更小子塊，再分派給空閒 Rank，加速 Level 2

預期輸出

- 是否有錯： `PASS | FAIL`
- 第一個錯位索引： `index = 600 000 534`
- 錯位區段摘要： `[600 000 534 ... 600 000 540]`
- 統計：掃描元素數、各階段耗時
- 可選 **Checksum**：每塊 `CRC32`，區分資料毀損 vs 排序錯誤

那我想說的是：

我想用這個專案展示，即使「只是檢查排序」，也能透過精心設計的平行流程，變成一個有深度、有應用價值的 HPC 範例。