arXiv:2603.00898v1 [cs.DS] 1 Mar 2026

高概率工作高效并行算法

Chase Hutton
马里兰大学 美国

Adam
Melrod 哈佛大学
美国

摘要

许多基础问题的随机化并行算法能够以期望达到最优的线性工作量，但要将这一保证提升为以高概率（以高概率）成立，仍是一项反复出现的理论挑战。本文针对若干核心并行原语弥补了这一差距。

首先，我们提出首个并行半排序算法，以$O(n)$工作量 以高概率 和 $O(\text{polylog}\, n)$深度 以高概率实现保证，改进了 Gu 等人 [16] 给出的$O(n)$期望工作量界。我们的分析引入了基于简单制表哈希的全新集中性论证，以及针对几何随机变量加权和的尾界。作为推论，我们得到一种针对$[n]$ 中键的整数排序算法，并达到相同界限。

其次，我们提出一个通用框架，将随机化并行图算法的工作量从期望线性提升至以高概率线性。该框架适用于局部可扩展问题——这类问题能够通过一个确定性过程，以与图割大小成正比的工作量，将解扩展到图割的另一侧。我们将其与剔除后的平衡划分方案相结合：迭代剔除阶段会移除多对数个高阶顶点；随后，通过有界差分论证，剩余图能够以高概率 进行平衡随机顶点划分。对小片段应用工作量低效但以高概率成功的子程序，并在割两侧进行确定性扩展，即可实现以高概率达到总体线性工作量。我们将该框架应用于构造工作量为 $O(m)$、以高概率 成功且深度为多对数级、以高概率 成功的算法，用于 $(\Delta + 1)$-顶点着色和极大独立集。

1 引言

随机化并行算法是现代算法设计的基石，能够为各种各样的问题提供简洁优雅的解法。在工作-深度模型中， [18], 并行算法的效率由其工作量（操作总数）和深度（最长的连续依赖链）衡量。理想情况下，并行算法的工作量应与最佳顺序时间复杂度相当，而深度应为多对数级，从而实现高效的并行执行。对于许多基础问题，随机化为实现这些目标提供了一条颇具吸引力的途径：Luby 极大独立集算法（MIS） [21] 和简单的随机化着色算法 [17, 19, 22] 等算法，都有工作量最优（线性工作量）、深度为多对数级的并行实现。

然而，对于许多这类算法，线性工作量的保证仅在期望意义下成立。期望线性工作量是一项有用的保证，但仍无法排除偶尔出现代价高昂的执行。更强也更理想的保证是，工作量为 $O(n)$（对于图问题，则为 $O(m)$），且以高概率（*whp*）成立，也就是说，成立概率为 $1 - n^{-\Omega(1)}$。将期望工作量界提升为高概率工作量界，是一项反复出现的理论挑战：使这些算法变得简单的随机性来源——随机抽样、随机哈希——往往会引入依赖关系和方差，而在整个执行过程中，这些因素很难得到严格控制。

半排序。 半排序是这样一个问题：重新排列包含 n 记录的数组，使键相同的记录彼此相邻，而无需按任何特定顺序排列不同键。作为排序的一种松弛形式，半排序满足了按键对记录分组这一常见需求，因此广泛应用于并行算法的设计与分析中 [1–5, 9–14, 20, 25–28]。Gu、Shun、Sun 和 Blelloch [16] 提出了一种自顶向下并行半排序算法，其期望工作量为 $O(n)$，深度为 $O(\log n)$ 以高概率。尽管这种算法很有用，但此前没有算法能以高概率实现 $O(n)$工作量。

图问题。 几种经典并行图算法也呈现出期望工作量界与高概率工作量界之间的类似差距。Luby 著名的最大独立集算法 [21] 可实现 $O(m)$的期望工作量和 $O(\log^2 n)$的深度以高概率。对于$(\Delta+1)$-顶点着色，一种广为人知的随机化算法——每个尚未着色的顶点独立地从可用颜色中均匀随机选取一种颜色，若没有邻居选中相同颜色则保留该颜色——可实现$O(m)$的期望工作量和 $O(\log n)$的深度以高概率。据我们所知，此前没有工作为$(\Delta + 1)$-着色证明高概率线性工作量界。

我们的贡献。 本文解决了若干基础并行原语的期望工作量界与高概率工作量界之间的差距。我们提出了首个并行半排序算法，其工作量为 $O(n)$以高概率 ，深度为 $O(\text{polylog}\, n)$以高概率。我们的算法遵循 Gu 等人 [16], 的自顶向下框架，但引入了新的集中性论证——利用简单的制表哈希以及几何随机变量加权和的尾界，将工作量界从期望提升至高概率。作为直接推论，我们得到了一种适用于 $[n]$ 中键的不稳定整数排序算法，其工作量为 $O(n)$以高概率 ，深度为 $O(\text{polylog}\, n)$以高概率。

其次，我们介绍一种框架，可将随机化并行图算法的工作量从期望线性提升至以高概率线性。该框架适用于具有自然的确定性可扩展性这一性质的问题：给定子图上的一个解，可以通过工作量与割大小成正比的方式，确定性地将其扩展过一条割。我们将此与剔除后的平衡划分方案结合，通过迭代顶点筛选和随机划分，将图分解为平衡的小片段。通过对小片段应用工作量不高效但以高概率成功的算法，并对割使用确定性扩展器，整体工作量以高概率仍保持线性。我们应用这一框架，据我们所知，得到了首个用于 $(\Delta+1)$-着色的 $O(m)$ 以高概率工作量算法，其深度为 $O(\text{polylog}\ n)$ 以高概率。此外，我们还基于 Luby 的算法，给出了一个用于最大独立集的 $O(m)$ 以高概率工作量、 $O(\text{polylog}\, n)$ 深度以高概率算法

算法。我们注意到，Blelloch 等人 [7]此前通过一种不同的方法，已为最大独立集实现了这一点。

2 预备知识

2.1 模型与基本操作

我们使用支持任意分叉的工作-深度模型来分析并行算法 [18]。一组线程在共享内存上运行，每个线程都支持标准 RAM 操作，以及一种可生成子线程的 分叉 指令。内存可以分配在栈上（仅分配该内存的线程可访问），也可以分配在堆上（可共享）。算法的工作量 是执行的指令总数，而深度 是最长依赖指令链的长度。

我们的算法使用以下标准基本操作，每种操作都作用于一个由 n 个元素组成的数组：

- **扫描**。 给定一个数组 A 和一个结合算子 $\oplus$，扫描会计算 A 在 $\oplus$下的所有前缀和序列。该操作需要 $O(|A|)$ 工作量（假设 $\oplus$需要 $O(1)$ 工作量）和 $O(\log |A|)$ 深度。
- **归约**。 给定数组 A 以及一个满足结合律的二元函数 f，归约计算将 f应用于 A的所有元素。这需要 $O(|A|)$工作量（假设 f需要 $O(1)$工作量）以及 $O(\log |A|)$深度。
- **划分**。 给定数组 A 以及谓词 P，划分会重排 A，使所有满足 P的元素出现在不满足的元素之前。利用扫描，这需要 $O(|A|)$工作量以及 $O(\log |A|)$深度。
- **过滤**。 给定数组 A 以及谓词 P，过滤会生成一个新数组，其中恰好包含使$P(a)$成立的那些 $a \in A$，并保持其原有顺序。这需要$O(|A|)$工作量以及 $O(\log |A|)$深度。
- **比较排序**。 给定由可比较元素组成的数组 A，比较排序会生成A的一个有序排列。使用并行归并排序 [8],，这需要 $O(|A|\log |A|)$工作量以及 $O(\log |A|)$深度。

2.2 集中界

2.2.1 Chernoff 界。 我们回顾独立伯努利变量之和的标准 Chernoff 界。

定理 2.1。 设 $X = \sum_{i=1}^{n} X_i$，其中 $X_i, i = 1, \ldots, n$ 是取值于 $\{0, 1\}$ 的独立随机变量。令 $\mu = \mathbf{E}[X]$。那么对于$\delta \geq 0$,

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right),$$

对于 $\delta \in [0, 1]$,

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2}\right).$$

2.2.2 几何随机变量之和。 我们陈述两个关于独立同分布的几何随机变量之和的集中性结果。

定理 2.2. 设 $G_1, \ldots, G_r$相互独立，且服从 $G_i \sim \mathrm{Ge}(p)$，其中某个 $p \in (0, 1]$,，并设 $G = \sum_{i=1}^{r} G_i$。那么对每个 $\lambda \geq 1$,

$$\Pr[G \geq \lambda \cdot \mathbf{E}[G]] \leq \exp\left(-\frac{(\lambda-1)^2}{2\lambda}\, r\right).$$

证明。设 $t := \lambda\mathbf{E}[G] = \lambda r/p$。考虑 t次相互独立的 Bernoulli(p) 试验，并令 $Y \sim \mathrm{Bin}(t, p)$ 表示成功次数。事件 $\{G > t\}$ 等价于 $\{Y < r\}$。由于 $\mathbf{E}[Y] = tp = \lambda r$，我们有

$$\Pr[G > t] = \Pr[Y < r] = \Pr[Y \leq (1-\delta)\mathbf{E}[Y]],$$

其中 $\delta := 1 - 1/\lambda \in [0, 1)$。Chernoff 下尾界（定理 2.1）给出 $\Pr[Y \leq (1-\delta)\mathbf{E}[Y]] \leq \exp(-\delta^2\lambda r/2) = \exp(-(\lambda-1)^2 r/(2\lambda))$。 □

定理 2.3。 令 $G_1, \ldots, G_r$ 相互独立，且服从 $G_i \sim$几何分布 $(1/2)$。令 $w_1, \ldots, w_r \geq 0$，并定义

$$W_1 := \sum_{i=1}^{r} w_i, \qquad W_2 := \sum_{i=1}^{r} w_i^2, \qquad W_\infty := \max_{i \in [r]} w_i.$$

那么，对每个 $t \geq 0$,

$$\Pr\left[\sum_{i=1}^{r} w_i G_i \geq 2W_1 + t\right] \leq \exp\left(-\min\left\{\frac{t^2}{16W_2}, \frac{t}{8W_\infty}\right\}\right).$$

证明。对于 $G \sim \mathrm{Ge}(1/2)$ 和 $\theta < \log 2$，$\mathbf{E}[e^{\theta G}] = \frac{1}{2}e^{\theta}/(1 - \frac{1}{2}e^{\theta})$。令 $\psi(\theta) := \log \mathbf{E}[e^{\theta(G-2)}] = \log \mathbf{E}[e^{\theta G}] - 2\theta$。可以验证，对于 $\theta \in [0, 1/4]$，$\psi(\theta) \leq 4\theta^2$。

令 Y_i: $= w_i\ G_i - 2$ $\sum_i w_i G_i = 2W_1 + (\ )$ Y $c \in [0, 1\ \ 4W_\infty]$ $cw_i \leq 1\ 4,$ $[e^{cY_i}] =$ 和 $Y := \sum_i Y_i$，于是 . 对于 /( )，我们有 /，因此 $\mathbf{EE}[e^{(cw_i)(G_i-2)}] \leq \exp(4(cw_i)^2)$。由独立性，

$$\mathbf{E}[e^{cY}] = \prod_i \mathbf{E}[e^{cY_i}] \leq \exp(4c^2W_2).$$

由马尔可夫不等式可得 $\Pr[Y \geq t] \leq \exp(-ct + 4c^2W_2)$。取 $c := \min\{t/(8W_2), 1/(4W_\infty)\}$。若 $c = t/(8W_2)$，则指数为 $-t^2/(16W_2)$。否则 $c = 1/(4W_\infty)$ 且 $t > 2W_2/W_\infty$，从而 $-ct + 4c^2W_2 = -t/(4W_\infty) + W_2/(4W_\infty^2) \leq -t/(8W_\infty)$。 □

2.2.3 有界差分。

定义 2.1。 函数 $f(x_1, \ldots, x_n)$满足具有常数 d_i、$i \in [n]$的利普希茨性质，如果对于仅在第 i 个坐标上不同的任意 $|f(\mathbf{a}) - f(\mathbf{a}')| \leq d_i$，其中$\mathbf{a}$ 和$\mathbf{a}'$。

定理 2.4（McDiarmid [23]）。若 f满足具有常数 d_i 的利普希茨性质，且$\mathbf{X} = (X_1, \ldots, X_n)$ 是独立随机变量，则

$$\Pr[|f(\mathbf{X}) - \mathbf{E}f(\mathbf{X})| > t] \leq 2\exp\left(-\frac{2t^2}{\sum_{i \leq n} d_i^2}\right).$$

2.3 哈希

我们将来自全集 U的 n 个键映射到 $[m]$。

2.3.1 通用哈希。 如果对于每一对不同键 $x \neq y \in U$，$\Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \leq 1/m$，则一个族 $\mathrm{H} = \{h : U \to [m]\}$ 称为通用（或2-通用）。标准构造（例如 $h_{a,b}(x) = ((ax + b) \bmod p) \bmod m$，其中 p 是素数）使用 $O(\log |U|)$ 比特的随机性，并可在 $O(1)$ 时间内求值。

2.3.2 简单制表哈希。 固定一个常数 c 和一个字母表 Σ，并将每个键视为向量 $x = (x_1, \ldots, x_c) \in \Sigma^c$。简单制表哈希抽取 c 个相互独立的随机表 $T_1, \ldots, T_c : \Sigma \to \{0, 1\}^w$（其中 $w = \log_2 m$），并定义

$$h(x) = T_1[x_1] \oplus T_2[x_2] \oplus \cdots \oplus T_c[x_c].$$

求值需要 $O(c) = O(1)$ 次表查找和异或运算。

集中性。 尽管只使用了有限的随机性，简单制表法在哈希到桶时仍表现出 Chernoff 型集中性。

定理2.5（P˘atraşcu–Thorup [24]）。考虑使用简单表格法将 n 球哈希到 $m \geq n^{1-1/(2c)}$ 桶中，每个键由 c 个字符组成。设 q为另一个查询球，并设 X_q 为哈希到某个桶中的普通球数量，该桶根据 h（q）选择。设 $\mu = \mathrm{E}[X_q] = n/m$。则对于任意常数 $\gamma > 0$:

$$\text{for all } \delta \leq 1: \quad \Pr\left[|X_q - \mu| > \delta\mu\right] < 2\exp\left(-\Omega(\delta^2\mu)\right) + m^{-\gamma}, \tag{2.1}$$

$$\text{for all } \delta = \Omega(1): \quad \Pr\left[X_q > (1+\delta)\mu\right] < (1+\delta)^{-\Omega((1+\delta)\mu)} + m^{-\gamma}. \tag{2.2}$$

此外，对于任意 $m \leq n^{1-1/(2c)}$，以至少 $1 - n^{-\gamma}$的概率，每个桶接收 $n/m \pm O$（$\sqrt{}$（n/m）$\overline{\text{对数}^c\, n}$）个球。

2.4 放置问题

放置问题接收一个包含 k条记录的输入数组 A，每条记录都携带一个属于 $[m]$ 的键。对于每个 $i \in [m]$，都有一个目标数组 B_i，其容量满足 $|B_i| \geq \alpha \cdot b_i$，其中 $b_i := |\{x \in A : \text{键}(x) = i\}|$，并且 b_i是 $\alpha \geq 2$ 一个常数松弛因子。目标是将键为 i的每条记录放入 B_i中的不同槽位。

算法。 将 A 划分为大小为 d 的块（最后一个块的大小至多为 $2d$）。每一轮中，每个块选择一个当前尚未放置的记录，并在其目标 B_i 中探测一个均匀随机槽位；如果该槽位为空，则插入该记录。此操作可以通过比较并交换操作完成。

引理 2.1。 固定一个大小至多为 $2d$ 的块，并令 R 表示放置该块中所有记录所需的轮数。则对每个 $\lambda \geq 1$,

$$\Pr\left[R > \lambda \cdot \frac{2d}{p}\right] \leq \exp\left(-\frac{(\lambda-1)^2}{2\lambda} \cdot 2d\right) = \exp(-\Omega(\lambda d)).$$

证明。在任意时刻， B_i 中至多有 b_i 个槽位被占用，因此一次随机探测成功的概率至少为 $p := 1 - 1/\alpha \geq 1/2$。用独立的 $\mathrm{Ge}(p)$ 随机变量耦合每条记录的尝试次数，并将定理 2.2 应用于至多 $2d$ 个此类随机变量之和。□

推论 2.1。 当 $d = \Theta(\log n)$ 时，放置算法以至少 $1 - n^{-\Omega(1)}$ 的概率在 $O(\log n)$ 轮内完成。每轮确定性地进行 $O(k/d)$ 次探测，因此总工作量为 $O(k)$ 以高概率。

2.5 图算法

我们介绍图 $G = (V, E)$ 上的两种并行图算法，其中有 $n = |V|$和 $m = |E|$。每种算法的工作量均为 $O(m \log n)$ 以高概率，深度为 $O(\log^2 n)$ 以高概率。在第 4 节中，我们将改进这两种算法，使其工作量达到 $O(m)$ 以高概率，同时保持多对数深度。

极大独立集。 Luby 算法 [21] 按轮次进行。在每一轮中，每个尚未移除的顶点 v 独立地随机选取一个值 $r(v)$，如果 $r(v)$ 是其尚未移除的邻居中的局部最大值，则 v 将其加入独立集。随后移除所有新加入的顶点及其邻居。由于每一轮平均会消除剩余边中的一个常数比例，因此只需 $O(\log n)$ 轮即可以高概率完成。每一轮都会扫描所有剩余的边一次，因此总工作量为 $O(m \log n)$ 以高概率，深度为 $O(\log^2 n)$ 以高概率。

$(\Delta+1)$-顶点着色。我们采用文献 [17] 中分析过的经典调色板采样算法。该算法以轮次为单位运行。令 S表示当前尚未着色的顶点集合，初始为 $S = V$。每一轮中，每个顶点 $v \in S$独立地从其调色板 P_v 中均匀随机采样一种颜色 $c(v)$；调色板是指在 $[\Delta+1]$ 中未被 v 的任何已着色邻居使用的颜色集合。由于每个顶点的调色板中至少有 $\deg_S(v)$ 种颜色（其中 $\deg_S(v)$ 表示未着色邻居的数量），而竞争者至多有 $\deg_S(v)$ 个，因此每个顶点都有常数概率成功。如果 v采样的颜色与某个邻居 $u \in S$ 冲突（即 $c(v) = c(u)$），两个顶点都会放弃所选颜色。否则， v着色并从 S 中移除。经过 $O(\log n)$ 轮后， S 以高概率变为空集。每轮的工作量为 $O(m)$（扫描所有剩余的边），深度为 $O(\log n)$，因此总工作量为 $O(m \log n)$，以高概率成立；深度为 $O(\log^2 n)$，也以高概率成立。

3 半排序

半排序 问题要求对包含 n 条记录的数组进行置换，使键相同的记录连续排列。以往的半排序并行算法要么实现期望工作量为 $O(n)$、深度为 $O(\log n)$ [6, 16] 的算法，要么实现工作量为 $O(n)$、深度为 $O(n^{3/4})$ [15]的算法（以高概率 ）。我们给出了首个工作量为 $O(n)$（以高概率 ）且同时保持多对数深度的算法。

3.1 概述

我们的算法采用 Gu–Shun–Sun–Blelloch 的自顶向下半排序框架 [16]（下文简称 GSSB），该框架实现了期望工作量为 $O(n)$、深度为 $O(\log n)$ 的算法（以高概率）。

3.1.1 GSSB 框架。 设 A为一个输入数组，其中包含 n条记录，其键取自任意全集 U。GSSB 算法由三个主要阶段组成。

阶段 *1*：采样与分桶。 以 $p_s = \Theta(1/\log n)$的概率独立采样每条记录，并将样本放入样本数组 S。对样本进行排序，以计算 A 中每个键 x对应的近似计数 σ_x，即键 x在 A中的重数。若键 x满足阈值 $\tau = \Theta(\log n)$的条件 $\sigma_x \geq \tau$，则将其标记为重型；其余记录为轻型。将重型和轻型记录分别划分到子数组 H和 L中。对于轻型记录，使用哈希函数 $h: U \to [B]$ 将每条记录分配到 $B = \Theta(n/\log^2 n)$个轻桶之一。

阶段 *2*：放置。对于每个重键 x，都会分配一个目标数组 D_x^H，其大小根据样本计数 σ_x 确定。同样，对于每个轻桶 $b \in [B]$，都会分配一个目标数组 D_b^L，其大小根据映射到 b 的轻型记录的采样数量确定。具体而言，如果 σ_x（分别为 σ_b）表示该计数，则目标数组的大小设为 $\alpha f(\sigma_x)$（分别为 $\alpha f(\sigma_b)$），其中 f 是精心选取的分配函数（定义见 (3.1)），而 $\alpha \geq 2$ 是一个常数松弛因子。随后，使用第 2.4 节所述的随机放置过程，将记录分发到各自的目标数组中。

阶段 *3*：局部半排序。放置后，每个轻型目标数组 D_b^L 都会被打包成一个大小为 m_b 的连续数组 C_b。现在的任务是对每个 C_b 内的记录进行半排序。在 GSSB 算法中，这是通过将 C_b 中的记录插入

并行哈希表，然后根据哈希值使用确定性基数排序进行排序来实现的。由于所用的并行哈希表支持每次操作的期望工作量为 $O(1)$，因此每个桶的期望工作量为 $O(m_b)$，总工作量的期望值为 $O(n)$。

3.1.2 我们的方法。使用标准论证，我们证明 GSSB 算法的阶段 1 和阶段 2 可在 $O(n)$ 工作量内完成，且以高概率成立 *whp*。因此，我们的主要技术贡献在于将阶段 3 的工作量上界从期望提升到高概率。我们分两步实现这一点。

步骤 *1*。第一步是推导打包轻桶 C_b大小的紧确上界。由于哈希到桶 b中的每个不同记录的重数至多为 $O(\log^2 n)$，且以高概率成立 *whp*，因此只需界定映射到 b的不同记录数量。我们将至多 n个不同记录哈希到 $B = \Theta(n/\log^2 n)$ 个桶中，因此每个桶中的不同记录期望数量为 $O(\log^2 n)$。在完全随机的哈希函数下，标准的球入箱集中性可立即给出以高概率成立的 $O(\log^2 n)$ 上界。然而，表示真正随机的哈希函数需要 $\Omega(n \log |U|)$ 位，因此并不实用。

我们发现，简单制表哈希已经足够。第 2.3.2 节介绍的简单制表哈希将每个键视为由 $c = O(1)$ 个字符组成的向量 $x = (x_1, \ldots, x_c) \in \Sigma^c$，并使用 c 个相互独立的随机查找表 $T_i : \Sigma \to \{0,1\}^w$，计算出 $h(x) = T_1[x_1] \oplus T_2[x_2] \oplus \cdots \oplus T_c[x_c]$，总共使用 $O(|\Sigma| \cdot c)$ 个随机字。其求值时间为 $O(1)$。P˘atraşcu 和 Thorup [24]证明，简单制表具有类似 Chernoff 界的逐桶集中性：将 n 个球哈希到 $m \geq n^{1-1/(2c)}$ 个桶中时，任意固定桶的负载满足

$$\Pr[X > (1+\delta)\mu] < (1+\delta)^{-\Omega((1+\delta)\mu)} + m^{-\gamma}$$

for $\delta = \Omega(1)$，其中 $\mu = n/m$ 是期望负载。在我们的情形中，$\mu = \Theta(\log^2 n)$，因此对于任意固定桶 b，应用此尾界可得失败概率为 $\exp(-\Omega(\log^2 n)) + B^{-\gamma}$。接着，对所有 B 个桶应用并集界，可得高概率下没有任何桶会接收超过 $O(\log^2 n)$ 条不同记录，因此 $\max_b |C_b| = O(\log^4 n)$ 也以高概率成立。

步骤 *2*。第二步替换 GSSB 在局部半排序阶段使用的并行哈希表。具体做法如下：在每个包含 C_b 个记录（其中至多有 $D \leq m_b$ 个不同键）的轻桶 m_b 中，我们从一个 2-通用族中采样一个哈希函数 $g : U \to [m_b^K]$（其中 $K \geq 3$ 为常数），并使用确定性基数排序，按照记录 C_b 的哈希值对其进行排序。如果 g 在 D 个不同键之间没有产生碰撞，那么根据通用性和并集界，这种情况发生的概率至少为 $1 - \binom{D}{2}/m_b^K \geq 1/2$，此时排序会生成 C_b 的有效半排序。线性时间扫描会检测碰撞；若发现碰撞，就重新采样 g 并重复上述过程。

我们注意到，桶 b 中的重新哈希尝试次数在随机意义上被几何随机变量 $X_b \sim \mathrm{Ge}(1/2)$支配，而桶 b中的总工作量为 $O(m_b \cdot X_b)$。重要的是，由于每次尝试都使用独立采样的哈希函数，因此随机变量 $\{X_b\}_{b\in[B]}$ 在各个桶之间相互独立。因此，阶段 3 的总工作量有如下上界：

$$W = O\left(\sum_{b\in[B]} m_b \cdot X_b\right),$$

一组相互独立的 Ge(1/2) 随机变量的加权和。为了对 W 进行界定，我们使用定理 2.3，它给出了 i.i.d. 几何随机变量加权和的集中不等式。具体而言，我们应用定理 2.2，其中 $W_1 = \sum_b m_b = |L|$、$W_2 = \sum_b m_b^2$、$W_\infty = \max_b m_b$ 和 $t = |L|$，得到：

$$\Pr[W \geq 3|L|] \leq \exp\left(-\min\left\{\frac{|L|^2}{16W_2}, \frac{|L|}{8W_\infty}\right\}\right).$$

注意，根据步骤 1，我们有 $W_\infty = O(\log^4 n)$ 以高概率。此外，我们可以将 W_2 界定为 $W_2 \leq W_\infty \cdot W_1 = O(|L|\log^4 n)$ 以高概率。因此，假设 $|L| \geq n/\log n$，可得

$$\Pr[W \geq 3|L|] \leq \exp\left(-\Omega\left(\frac{|L|}{\log^4 n}\right)\right) = n^{-\omega(1)},$$

从而以高概率得到 $W = O(|L|) = O(n)$。

3.1.3 结果与组织结构。 我们在如下定理中陈述主要结果。

定理 3.1。 存在一种并行半排序：给定一个由 n条记录组成的数组，它使用 $O(n)$工作量和 $O(\text{多重对数 } n)$深度以高概率产生半排序输出。

本节其余部分组织如下。第 3.2 节给出完整算法。第 3.2.2 节建立采样与分配保证，以确保阶段 1 和阶段 2 以线性工作量运行。第 3.2.3 节进行桶大小分析，并给出阶段 3 的高概率工作量界，这是论证的核心。最后，第 3.2.4 节考虑重新启动的可能性，并完成定理 3.1 的证明。

3.2 算法

回顾一下，A 是一个包含 n 条记录的数组，其键属于全集 U。该过程见算法 1。

3.2.1 参数与符号。固定常数 $K \geq 3$、$\alpha \geq 2$和$c > 0$。令 $p_s = \Theta(1/\log n)$为采样概率，$\Theta(\log n)$为重桶阈值。令

$$B := \lceil \log_2(n/\log^2 n) \rceil = \Theta\left(\frac{n}{\log^2 n}\right)$$

为轻桶的数量。符号汇总见表 1。我们还回顾 GSSB 算法使用的分配函数。对于 $s \geq 0$，定义

$$f(s) := \frac{s + c\log n + \sqrt{c^2\log^2 n + 2sc\log n}}{p_s}. \tag{3.1}$$

3.2.2 采样与分配保证。本节将验证采样步骤能准确近似输入分布，并且第 2 阶段分配的目标数组同时足够大，能够容纳每个键对应的所有记录；同时其总空间又足够小，从而保证 $O(n)$空间。

由于 n 条记录中的每一条都以 $p_s = \Theta(1/\log n)$的概率独立纳入，因此 $|S| \sim$服从 $\mathrm{Bin}(n, p_s)$，由 Chernoff界（定理 2.1）可立即得到：以 $1 - \exp(-\Omega(n/\log n))$的概率，$|S| = \Theta(n/\log n)$。

下面的引理表明，如果 $W \sim \mathrm{Bin}(\widehat{W}, p_s)$是观测到的样本计数，那么 $f(\widehat{W}) \geq W$以高概率成立。

符号	含义
A, n	输入数组和记录数
H, L	重记录子数组和轻记录子数组
p_s	采样概率，$\Theta(1/\log n)$
S	采样记录数组
σ_x, σ_b	S 中键 x（相应地，桶 b）的样本计数
τ	σ_x, Θ 上的重桶阈值（$\log\ n$）
h	简单制表哈希函数 $U \to [B]$
B	轻桶数量，Θ（$n/\log^2 n$）
桶$(\cdot)$	桶索引映射 $L \to [B]$ 经由 h
$f(\cdot), \alpha$	分配函数和松弛因子
D_x^H	键 x的重型目标数组
D_b^L	桶 b的轻型目标数组
C_b, m_b	打包的轻桶 b 及其大小 $\lvert C_b\rvert$

表 1：第 3.2 节的记号。

引理 3.1 $W \geq 0$ W p_s [f <.. 设和 $\widehat{W} \sim$ $W] \leq n^{-c}$ Bin(,)。则Pr($\widehat{W}$)。

证明。记 $\mu := Wp_s = \mathbf{E}[\widehat{W}]$。由于 f是非递减的，事件 $\widehat{\ }\{f(W) < W\}$ 等价于 $\{W \leq s\}$，其中 s是满足 $f(s) < W$ 的最大整数。令 $t := \mu - s > 0$。不等式 $W > f(s)$ 给出 $\mu > s + c$ $^2\overline{n + 2sc\quad n\quad t - c}\quad n^{\,2} > \log\ n + \sqrt{c^2}$ $c^2\quad ^2n{+}2sc\quad n\qquad t^2 > 2c\quad n$ log log，因此 (log) log log，其化简为 log $\cdot\mu$。Chernoff 下尾界（定理 2.1）进而给出

$$\Pr[\widehat{W} \leq s] = \Pr[\widehat{W} \leq \mu - t] \leq \exp\left(-\frac{t^2}{2\mu}\right) = \exp(-c\log n) = n^{-c}.$$

□

通过联合界将该引理应用于所有键和桶，我们可以控制任意目标数组出现分配不足的概率。令 N_x 表示键 x = $\lvert\ x \in A$ $x\ = b\ \rvert$ 的真实重数，N_b : { : bucket( ) } 表示真实桶大小。定义分配不足事件

$$E_{\text{under}} := \big(\exists x \in A : N_x > f(\sigma_x)\big) \vee \big(\exists b \in [B] : N_b > f(\sigma_b)\big).$$

对于每个键 x，样本计数 $\sigma_x \sim \text{Bin}(N_x,\ p_s)$，因此由引理 3.1 可得 $\Pr[N_x > f(\sigma_x)] \leq n^{-c}$；对至多 n 个键应用并集界，得到失败概率至多为 n^{1-c}。对每个 $B \leq n$ 个桶应用相同的论证，其中 $\sigma_b \sim \text{Bin}(N_b,\ p_s)$，另贡献 n^{1-c}。由此可得以下引理。

Lemma 3.2. $\Pr[E_{\text{under}}] \leq O(n^{1-c})$.

还需验证总分配空间为 $O(n)$。令 $\mathrm{I} := H \cup [B]$，并定义

$$T := \alpha\left(\sum_{x\in H} f(\sigma_x) + \sum_{b\in[B]} f(\sigma_b)\right). \tag{3.2}$$

引理 3.3。 $T = \Theta(n)$，其成立概率为 $1 - n^{-\Omega(1)}$。

证明。对于所有 $s \geq 0$，不等式 $\sqrt{c^2\log^2 n + 2sc\log n} \leq c\log\ n + \sqrt{2sc\log n}$给出

$$p_s f(s) \leq s + 2c\log n + \sqrt{2sc\log n}. \tag{3.3}$$

Algorithm1:并行半排序

输入： 包含 n 条记录的数组 A，其中的键属于 U。

输出： 按半排序顺序排列的数组 A'。1. **采样**。以概率 p_s 独立采样每条记录 $x \in A$，并将每条采样到的记录 x 存入数组 S。2. **样本计数**。对 S 排序，并计算其中每个键 x 的重数 σ_x。3. **重轻划分**。令 $H := \{x\colon\ \sigma_x \geq \tau\}$，且 $L := A \setminus H$。4. **重键放置**。(a) 如果 $|H| < n/\log n$，则对 H进行比较排序，并将其存为一个段。(b) 否则，对每个键 $x \in H$ 分配大小为 $\alpha f(\sigma_x)$ 的空间 D_x^H。使用放置算法（第 2.4 节），以块大小 $d = \Theta(\log\ n)$ 将每条记录 $x \in H$ 放入 D_x^H。如果放置未能在 $\Theta(\log\ n)$ 轮内完成，则转到步骤 1。

5. **轻键放置**。
   (a) 如果 $|L| < n/\log n$，则对 L 进行比较排序，并将其存储为一个段。(b) 否则，计算简单制表哈希 h : $U \to [B]$，并为每个 $x \in L$设置 bucket(x) := $h(x)$。计算桶样本计数$\sigma_b := |\{y \in S \cap L : \text{bucket}(y)\ = b\}|$（针对每个 $b \in [B]$），分配大小为 $\alpha f(\sigma_b)$ 的 D_b^L，并使用放置算法及 $d = \Theta(\log\ n)$ 将每条记录$x \in L$放入 D^L 桶 (x)。如果放置算法在 $\Theta(\log n)$ 轮内未完成，则转到步骤 1。
6. **局部半排序**。对每个轻桶 b 并行执行：通过过滤将 D_b^L 打包到连续数组 C_b 中，并令 $m_b := |C_b|$。然后：
   (a) 采样一个二通用哈希 $g\colon\ U \to [m_b^K]$。(b) 按哈希值对 C_b进行基数排序。(c) 扫描相邻记录：如果存在一对哈希值相同但键不同的记录（发生碰撞），则转到 (a)。
7. **打包**。使用前缀和将所有段打包到连续输出数组 A' 中，并返回 A'。

我们分别界定 $\sum_{i\in I}\ f(\sigma_i)$ 的重部分和轻部分。

对于重部分，应用 (3.3)，并利用 $\sum_{x\in H}\sigma_x \leq |S|$, $|H| \leq$ $|S|\ \tau$ $\sum_{x\in H}\sqrt{\sigma_x} \leq \sqrt{|H|}\ \ \sum_x \sigma_x \leq$ / 和柯西-施瓦茨界 · /, 可 $|S|\ \sqrt{\tau}$ $p_s\sum_{x\in H} f\ \sigma_x\ = O\ |S|$ 得 ( ) ( )。

对于轻部分，使用 $\sum_b \sigma_b \leq |S|$ 和 $\sum_b \sqrt{\sigma_b} \leq \sqrt{B|S|}$ 进行同样的论证，可得

$$p_s \sum_{b\in[B]} f(\sigma_b) \leq |S| + 2c\log n \cdot B + \sqrt{2c\log n \cdot B|S|}.$$

代入 $B = \Theta(n/\log^2 n)$ 和 $|S| = \Theta(n/\log\ n)$ 后，右侧为 $\Theta(n/\log\ n)$ 以高概率。合并并除以 $p_s = \Theta(1/\log\ n)$，得到 $\sum_{i\in I}\ f(\sigma_i)\ = \Theta(n)$，因此 $T = \Theta(n)$ 以高概率。□

3.2.3 轻桶分析与局部半排序。现在我们进入论证的核心：证明阶段 3（即对轻桶进行局部半排序）以高概率在 $O(n)$ 总工作量内完成。

分析分为两个阶段。首先，我们证明每个紧凑桶 C_b 的大小至多为 $O(\log^4 n)$。然后，我们利用这个界以及重新哈希方案带来的独立性，使总工作量集中。

桶大小。 紧凑桶的大小 C_b 受两个量的乘积限制：哈希到 b 的不同键的数量，以及这些键中任意一个键的最大重数。我们依次对这两个量进行界定，先从重数开始。

如果键 x 满足 $\mathbf{E}[\sigma_x] \geq 2\tau$，Chernoff界给出 $\Pr[\sigma_x < \tau] \leq \exp(-\Omega(\log n)) = n^{-\Omega(1)}$；对至多 n 个键应用并集界可知，以高概率 每个被归类为轻的键都满足 $N_x p_s = \mathbf{E}[\sigma_x] < 2\tau$，因此 $N_x = O(\log^2 n)$。

对于每个桶中不同键的数量，我们利用简单制表哈希具有 Chernoff 型集中性这一事实。令 n_* 表示 L 中的不同键数量，Y_b 表示映射到桶 b 的不同键数量。我们将初始哈希视为使用简单表格法，根据 c 个字符把 n_* 个球投入 B 个桶中。任意一个桶的预期负载为 $\mu = n_*/B = O(\log^2 n)$，并且对于足够大的 n，定理 2.5 的前提条件 $B \geq n_*^{1-1/(2c)}$ 得到满足。取 $t = \Theta(\log^2 n)$，使得 $\delta := t/\mu - 1 = \Omega(1)$，则尾界 (2.2) 给出

$$\Pr[Y_b > t] \leq (1+\delta)^{-\Omega(t)} + B^{-\gamma} = \exp(-\Omega(\log^2 n)) + B^{-\gamma}.$$

对 $B \leq n$ $\gamma \geq 3$ $_b Y_b =$ 个桶应用并集界，可得 O $^2 n$，因此 max (log) 以高概率。结合这两个界可得：

引理3.4。 假设 $|L| \geq n/\log n$。以概率 $1 - n^{-\Omega(1)}$，每个打包的轻桶都满足 $|C_b| = O(\log^4 n)$。

局部半排序的总工作量。 现在我们分析算法 1 的步骤 (e)(iv) 中的重新哈希循环。考虑桶 C_b 中的一次尝试，其中有 m_b 条记录，且其中 $D \leq m_b$ 条各不相同。我们使用 2-通用族将其哈希到大小为 m_b^K 的范围内，因此根据并集界，碰撞概率至多为

$$\binom{D}{2} \cdot \frac{1}{m_b^K} \leq \frac{m_b^2}{2m_b^K} = \frac{1}{2m_b^{K-2}} \leq \frac{1}{2}$$

对于 $K \geq 3$。因此，每次尝试都以至少 1/2 的概率独立成功，所以尝试次数在随机意义下被 $X_b \sim \mathrm{Ge}(1/2)$ 所支配。此外，变量 $\{X_b\}_{b\in[B]}$ 在各个桶之间相互独立。

桶 b 中的每次尝试需要 $O(m_b)$ 的工作量（哈希、基数排序以及用于处理碰撞的线性扫描），因此总工作量为

$$W = O\left(\sum_{b\in[B]} m_b X_b\right).$$

这是独立几何随机变量的加权和，我们可以利用定理 2.3 对其进行集中界定。以引理3.4中的事件为条件，因此 $W_\infty = \sum_b m_b m_b$ 且 O $^4 n$ $W_2 = \sum_b m_b^2 \leq W_\infty$ $W_1 = O$ $|L|$ $^4 n$ max (log)，且 · (log)。将定理应用于 $t = |L|$，得

$$\Pr\left[\sum_b m_b X_b \geq 3|L|\right] \leq \exp\left(-\Omega\left(\frac{|L|}{\log^4 n}\right)\right).$$

根据假设，$|L| \geq n/\log \omega$ n $n^{-\omega}$ 1 n $=$，指数为 $\Omega(/\log^5 n)$ (log)，因此失败概率为 ()。我们将此记录如下：

引理 3.5。 假设 $|L| \geq n/\log n$。步骤 *(e)(iv)* 的总工作量为 $O(|L|)$，其成立概率为 $1 - n^{-\omega}{}_{(1)}$。

3.2.4 重启分析与总体工作界。 每当放置阶段超过 $\Theta(\log n)$ 轮时，算法就会重新启动。现在我们证明，重新启动的概率足够低，并且所有迭代的总工作量为 $O(n)$ 以高概率。

令 E_H（分别为 E_L）表示重（分别为轻）放置未能在 $O(\log n)$ 轮内完成的事件。以 $\neg E_{\text{under}}$ 为条件，每个目标数组的容量至少是其必须接收的记录数的 α 倍，因此满足放置问题的松弛条件。于是推论 $n^{-\Omega(1)}$ E_L 2 1 $[E_H \mid \neg E_{\text{under}}] \leq$. 给出 Pr（轻放置的情况同理）。结合引理 .，可得：3 2,

引理 3.6. E_H 和 E_L 发生的概率均至多为 $n^{-\Omega(1)}$。

现在我们证明定理 3.1。

定理 3.1 的证明。根据引理 3.6，以高概率不会发生重新启动，因此只需在此假设下界定工作量和深度。

工作量。采样需要 $O(n)$ 工作量。使用并行归并排序对 S 排序，以高概率需要 $O(|S|\log|S|) = O(n)$ 工作量。重轻划分需要 $O(n)$。对于重型记录：若 $|H| < n/\log n$，则对 H 进行比较排序需要 $O(|H|\log|H|) = O(n)$ 工作量；否则，在事件 $\neg E_H$ 发生时，放置运行 $O(\log n)$ 轮，每轮进行 $O(|H|/\log n)$ 次探测，总计 $O(|H|)$。对于轻型记录：若 $|L| < n/\log n$，则对 L 进行比较排序需要 $O(|L|\log|L|) = O(n)$ 工作量；否则，在事件 $\neg E_L$ 发生时，根据相同的论证，放置需要 $O(|L|)$，而引理 3.5 以高概率将局部半排序的工作量界定为 $O(|L|)$。两种情况下，总计均为 $O(|H| + |L|) = O(n)$。根据引理 3.3，以高概率分配空间为 $T = \Theta(n)$，因此打包目标数组和组装最终输出各需要 $O(n)$。因此，以至少 $1 - n^{-\Omega(1)}$ 的概率，工作量至多为 $O(n)$ 工作量。

深度。采样、划分和最终打包步骤的深度均为 $O(\log n)$。排序 S 需要 $O(\log n)$ 深度。对于重型记录：如果 $|H| < n/\log n$，比较排序的深度为 $O(\log n)$；否则，放置需要进行 $O(\log n)$ 轮，每轮深度为 $O(\log n)$，总计贡献 $O(\log^2 n)$。对于轻型记录：如果 $|L| < n/\log n$，比较排序的深度为 $O(\log n)$；否则，放置贡献 $O(\log^2 n)$，而局部半排序占主导地位。在后一种情况下，对大小为 $m_b = O(\log^4 n)$ 的桶进行每次重新哈希尝试时，会执行一次基数排序，其中包含 $K = O(1)$ 轮以 m_b 为基数的计数排序，每轮深度为 $O(m_b)$，因此每次尝试的深度为 $O(\log^4 n)$。对于全部 B 个桶，尝试次数的最大值为 $\max_b X_b$，其中 $X_b \sim$ 独立地满足 Ge(1/2)；由并集界可得 Pr $[$ $_b X_b > \Omega$ n $] \leq$ max (log) B $2^{-\Omega(\log n)} \leq n^{1-\Omega(1)}$ $_b X_b = O$ n ·，因此以高概率 max (log)。因此，局部半排序的深度为 $O(\log^4 n \cdot \log n) = O(\log^5 n)$，以高概率成立，并且它超过所有其他步骤。□

3.3 应用：整数排序

作为直接应用，我们的半排序可得到针对整数键的不稳定计数排序。

推论 3.1. 给定数组 A，其中有 n 条键位于 $[n]$ 中的记录，A 的不稳定计数排序可在 $O(n)$ 工作量 以高概率 以及 $O($多重对数 $n)$ 深度 以高概率内计算得到。

证明。首先，对 A 应用定理 3.1，生成数组 A'，使得具有相同键的记录彼此相邻。接下来，扫描

A' 以识别组边界（键发生变化的位置），并将每组的大小写入计数数组 $C[0:\ n-1]$。最后，对 C 计算前缀和以获得输出偏移量，并将每组复制到输出数组中对应的区间。第一步占主导，因此工作量为 $O(n)$ 以高概率，深度为 $O(\text{polylog}\, n)$ 以高概率。□

4 图算法框架

我们的方法适用于展现特定确定性可扩展性质的图问题。为说明这一点，考虑将图 H 划分为不相交的子图 $H_0 \sqcup H_1$。假设我们已拥有该问题在 H_0 上的解，则需要一种确定性算法——其工作量与割的规模呈线性——以促成计算与 H_0 相容的 H_1 上的解。随后，回到 H_0 的次级桥接操作使得能够计算 H 的完整解。

我们方法的核心洞见是，某些图问题允许简单算法以 $O(m\,\text{polylog}\ n)$ 工作量以高概率运行，同时也允许这些确定性扩展器。借助这些工具，如果我们能产生将 G 划分为子图 $H_1, \ldots\ldots, H_{\text{polylog}\, n}$ 的平衡划分

，

平衡，使得 $|H_i| = O(m/(\text{polylog}\, n)^2)$，我们可以归纳地将工作量低效的以高概率 解应用于内部部分，并将确定性扩展器应用于割。该策略最终为整个图产生一个总体以高概率 线性工作量算法。

4.1 剔除后的平衡划分

一旦确定性扩展器的存在性得到确立，问题便归结为生成一个平衡划分。由于并非所有图都能以这种朴素方式划分，我们引入一种结构性放宽，称为剔除后的平衡划分。该划分由一小组被移除顶点 $C \subseteq V$（其中 $|C| \le \text{polylog}\, n$）以及剩余诱导子图 $G - C$ 的一个平衡划分 $\{V_1, \ldots, V_k\}$ 组成。形式化地说

定义 4.1。 大小为 k 的剔除后的平衡划分是一个元组 $(C, \{V_1, \ldots, V_k\})$，其中 $C \subseteq V$ 的大小为 $O(\text{polylog}\ n)$，且 $\{V_1, \ldots, V_k\}$ 是 $V \setminus C$ 的一个划分，使得对于所有 i，$G[V_i]$ 都有 $O(m/k^2)$ 条边。

为了构造大小为 polylogarithmic 的剔除后的平衡划分，我们采用有界差分法。该方法指出，若图中不存在度数过高的顶点，将顶点集随机划分为 $\text{polylog}\, m$ 个部分，就能以高概率实现边的平衡分布。因此，我们采用迭代剔除过程，不断剪除图，直到随机划分变得可行。

- **迭代剔除：** 我们迭代地识别并移除度较高的顶 C，并将其加入剔除集。如果不存在这样的顶点，我们就对剩余顶点进行随机划分。度条件保证单次迭代移除的顶点数至多为 $\text{polylog}\, n$。
- **进展保证：** 每轮之后，要么满足度条件（从而可以应用随机划分，以高概率实现平衡），要么剩余图中的边总数至少减少一半。
- **终止性：** 由于不成功轮中边数会减半，因此该迭代最多经过 $\log n$ 轮即可终止。

最后，为了计算整体解，我们处理平衡划分，并以确定性方式将解扩展到较小的剔除集 C。由于 C 的规模非常小，处理剔除部分所需的工作量对整体复杂度的贡献可以忽略不计，因此我们可以得出结论：整个算法具有严格的线性工作量界。

4.2 应用

4.2.1 $(\Delta+1)$-顶点着色。通过结合标准的调色板采样着色例程（将其作为工作量低效但以高概率成功的子程序）与跨割的确定性扩展器，我们获得了一个线性工作量、以高概率完成的 $(\Delta+1)$-着色算法。

确定性扩展器。设 H 划分为 $H = H_0 \sqcup H_1$，并假设 H_0 已由 ϕ 正确着色。对于每个顶点 $v \in V(H_1)$，定义其剩余调色板

$$C(v) := [\Delta+1] \setminus \{\phi(u) : u \in N_G(v) \cap V(H_0)\}.$$

计算这些调色板需要对每条割边恰好扫描一次，因此工作量为 $O(|E(H_0, H_1)|)$。此外，

$$|C(v)| \ge \deg_{H_1}(v) + 1 \qquad \text{for all } v \in V(H_1),$$

由于从大小为 $\Delta+1$ 的调色板中最多移除 $\Delta - \deg_{H_1}(v)$ 种颜色。因此，在 H_1 上使用调色板 $\{C(v)\}$ 的任何列表着色过程都会产生与 ϕ 兼容的正常着色。

算法。 设 $(C, \{V_1, \ldots, V_k\})$ 为 G 的一个剔除后的平衡划分，且具有 $k = O(\log n)$。对每个部分记 $H_i := G[V_i]$。算法如下进行。

(1) 对于 $i = 1, \ldots, k$：从已着色顶点 $H_1 \sqcup \cdots \sqcup H_{i-1}$ 向 H_i 应用确定性扩展器以获得剩余调色板，然后在 H_i 上用这些调色板运行调色板采样着色例程。(2) 使用确定性扩展器从 $H_1 \sqcup \cdots \sqcup H_k$ 扩展到剔除集 C，并用调色板采样过程对 C 着色。

工作量和跨度。 令 $m_i := |E(H_i)|$ 表示部分 i 的内部边数量，令 $c_i := |E(H_1 \sqcup \cdots \sqcup H_{i-1},\ H_i)|$ 表示进入 H_i 的割边数量。将 $W_{\text{ps}}(m')$ 和 $S_{\text{ps}}(m')$ 分别记为在具有 m' 条边的子图上运行调色板采样子程序的工作量和跨度；根据假设，$W_{\text{ps}}(m') = O(m' \log n)$ 以高概率。

处理部分 H_i 需要为扩展器付出 $O(c_i)$ 的工作量，着色需要 $O(W_{\text{ps}}(m_i))$ 的工作量。因此，总工作量为

$$\sum_{i=1}^{k} [O(c_i) + O(W_{\text{ps}}(m_i))] + O(W_{\text{ps}}(|C|)).$$

G 中的每条边至多是一个部分的割边，因此 $\sum_i c_i \le m$。由于该划分是平衡的，对于每个 i 以高概率，都有 $m_i = O(m/k^2)$，从而

$$\sum_{i=1}^{k} W_{\text{ps}}(m_i) = \sum_{i=1}^{k} O\left(\frac{m}{k^2} \cdot \log n\right) = O\left(\frac{m}{k} \cdot \log n\right) = O(m),$$

其中，最后一个等式使用了 $k = \log n$。剔除集满足 $|C| \le \text{polylog}\ n$，因此其贡献为 $O(\text{polylog}^2\, n) = o(m)$。所以，总工作量为 $O(m)$ 以高概率。

这些部分按顺序处理。部分 i 的扩展器以 $O(\log n)$ 跨度运行（对割边进行并行扫描），并且

调色板采样过程贡献 $S_{\text{ps}}(m_i)$跨度。因此，总体跨度为

$$\sum_{i=1}^{k}[O(\log n)+S_{\text{ps}}(m_i)]+O(\log n)+S_{\text{ps}}(|C|)$$
$$=O(k\cdot S_{\text{ps}}(m/k^2))+O(k\log n).$$

由于调色板采样过程可实现多对数跨度，因此可简化为 $O(\text{polylog}\, n)$。

4.2.2 极大独立集。结合 Luby 算法和确定性扩展器，我们得到了一个工作量为线性的、以高概率 运行的极大独立集算法。Luby 算法以 $O(m\log n)$ 的工作量以高概率运行；一旦描述了确定性扩展器，该算法及其分析便与顶点着色的算法及分析相同。

确定性扩展器。设 H划分为 $H=H_0\sqcup H_1$，并假设 M_0 是 H_0的一个极大独立集。我们并行删除每个顶点 $v\in V(H_1)$，只要它拥有一条连接到 M_0 中某个元素的割边，该顶点就从 H_1中删除。因此，计算更新后 H_1 的一个最大独立集 M_1，即可得到 H的一个最大独立集 $M:=M_0\cup M_1$。

4.3 通过剔除构造剔除后的平衡划分

我们分两个阶段构造剔除后的平衡划分。首先，剔除一小组高阶顶点，以满足某个最大度与边数之比条件；其次，一旦满足该条件，就将剩余顶点随机划分到 $k=\Theta(\text{polylog}\, n)$ 个桶中。该条件表明，这样做会得到一个剔除后的平衡划分。

4.3.1 阶段 I：剔除高阶顶点。我们首先初始化一个空的剔除集 $C\subseteq V$。剔除阶段分为若干阶段。在某个阶段中，令 H 表示剔除期间的当前图（初始为 $H\leftarrow G$）。记 $e(H)$ 为 H 中的边数，记 $\Delta(H)$ 为 H 的最大度。设置移除阈值

$$\tau(H):=\frac{e(H)}{k^4\log n}.$$

如果 $\Delta(H)>\tau$ 成立，则并行地从 H 中移除所有满足 $\deg_H(v)>\tau/2$ 的顶点 v，并将它们加入剔除集 C。如果该条件不成立，则剔除步骤结束，并令 $G':=G[V\setminus C]$ 为剩余图。

引理 4.1. 在以 M 条边开始的阶段结束时，要么$\Delta(H)\le e(H)/k^4\log n$成立，要么边数已减半：$e(H)\le M/2$。特别地，剔除过程需要 $O(m)$工作量和$O(\log n)$轮次。

证明。设 H'为该阶段结束时的图，其中 $M':=e(H')$，$\Delta':=\Delta(H')$，以及：()。若该阶段以/结束，则证明完成。否则 $M'>M/2$，且违反初始度条件的顶点已被移除，因此 $\Delta'\le\tau/2=M/(2k^4\log n)$。由于$M'>M/2$，我们有 $M<2M'$，因此

$$\Delta'\le\frac{M}{2k^4\log n}<\frac{2M'}{2k^4\log n}=\frac{M'}{k^4\log n}.$$

我们得到该引理 . □

我们还需要对 C 的大小给出一个界。

引理 4.2。在任意阶段，被移除顶点的数量至多为 $4k^4$对数 n。因此 $|C|=O($多重对数 $n)$，因为轮次数为 $O($对数 $n)$且 $k=O($多重对数$n)$。

证明。设 r 为被移除顶点的数量。每个被移除顶点的度至少为 $\tau/2$，因此移除所有这些顶点至少删除 $r\tau/4$条边（因为每条边至多与两个被移除顶点相邻）。被移除的边数不能超过边的总数，因此我们得出 $r\tau/4\le m$。这可化简为 $r\le 4k^4\log n$，即为所求。□

剔除结束时，我们得到 $G'=(V',E')$，其中 $V'=V\setminus C$,$n':=|V'|$, $m':=|E'|$，并且

$$\Delta(G')\le\frac{m'}{k^4\log n}.$$

4.3.2 第二阶段：对剩余顶点进行随机划分。给定$G'=(V',E')$，将 V'中的每个顶点独立且均匀地分配到 $[k]$中的一个桶中，得到划分 $V'=V_1\sqcup\cdots\sqcup V_k$，并输出 $(\{V_1,\dots,V_k\},C)$。对每个 $i\in[k]$，定义

$$X_i:=e(G'[V_i]).$$

根据期望的线性性，

$$\mathbf{E}[X_i]=\frac{m'}{k^2}.$$

引理 4.3。对于每个固定的 $i\in[k]$ 以及任意 $t>0$,

$$\Pr[|X_i-\mathbf{E}[X_i]|>t]\le 2\exp\left(-\frac{t^2}{\Delta(G')\,m'}\right).$$

证明。变量 X_i是标签 $(Z_v)_{v\in V'}$的函数 f_i，其中 $Z_v\in[k]$是 v所在的桶。改变单个顶点 v的标签，只会通过与 v关联的边影响 X_i，因此 f_i是利普希茨的，其常数为 $d_v\le\deg_{G'}(v)$。应用定理 2.4，令 $\sum_v d_v^2\le\sum_v\deg_{G'}(v)^2\le\Delta(G')\sum_v\deg_{G'}(v)=2\Delta(G')m'$，可得

$$\Pr[|X_i-\mathbf{E}[X_i]|>t]\le 2\exp\left(-\frac{2t^2}{2\Delta(G')m'}\right)=2\exp\left(-\frac{t^2}{\Delta(G')m'}\right),$$

正如所需。 □

将此引理应用于 G'，其中 $t=\Theta(m'/k^2)$，并利用$\Delta(G')\le m'/(k^4\log n)$，可得

$$\Pr\left[X_i=O\left(\frac{m}{k^2}\right)\right]\le 2\exp\left(-\Omega\left(\frac{k^4\log n}{m'}\frac{1}{m'}\frac{m'^2}{k^4}\right)\right)$$
$$\le 2\exp(-\Omega(\log n))=\frac{1}{n^{\Omega(1)}}.$$

随后通过并集界可得，对于所有 i，$X_i=O(m/k^2)$以高概率。执行该划分需要线性工作量和对数跨度，因此我们得到以下定理。

定理 4.1。对于任意多对数的 k，执行第一阶段和第二阶段会以线性工作量 以高概率 和多对数跨度生成剔除后的平衡划分 $(C,\{V_1,\dots,V_k\})$。

4.3.3 根据划分组织图。 计算出剔除后的平衡划分 $(C, \{V_1, \ldots, V_k\})$ 后，我们会重新组织图的表示，以便高效访问每个部分 $G[V_i]$ 以及部分之间的割边。我们对顶点数组按划分索引作为键应用推论 3.1 中的整数排序，使每个 V_i 的顶点占据一个连续段。接下来，对每个顶点 $v \in V_i$，我们再次使用整数排序，按邻居的划分索引对其邻接表排序。这样，每个邻接表都会被分成一个包含内部 邻居（即属于 V_i的邻居）的块，以及一个包含割边 邻居（即属于其他部分或 C的邻居）的块，并记录两者之间的边界作为分割点。整个物化过程需要 $O(m)$ 工作量 以高概率，深度为 $O(\log^5 n)$ 以高概率，其开销主要由整数排序决定。

参考文献

[1] U. A. Acar、D. Anderson、G. E. Blelloch 和 L. Dhulipala。并行批动态图连通性。载于 ACM算法与体系结构并行性研讨会 (SPAA)，第 381–392 页，2019 年。[2] U. A. Acar、D. Anderson、G. E. Blelloch、L. Dhulipala 和 S. Westrick。通过变更传播实现并行批动态树。载于 欧洲算法研讨会 (ESA)，2020 年。[3] D. Anderson、G. E. Blelloch 和 K. Tangwongsan。应用于滑动窗口模型的高工作效率批增量最小生成树。载于 ACM算法与体系结构并行性研讨会 (SPAA)，2020 年。[4] G. E. Blelloch、Y. Gu、J. Shun 和 Y. Sun。随机增量算法中的并行性。《ACM期刊》，67(5):1–27，2020 年。[5] G. E. Blelloch、Y. Gu、Y. Sun 和 K. Tangwongsan。使用半径步进求解并行最短路径。载于 ACM算法与体系结构并行性研讨会 (SPAA)，第 443–454 页，2016 年。[6] Guy E. Blelloch、Jeremy T. Fineman、Yan Gu 和 Yihan Sun。二叉分叉模型中的最优并行算法。载于第 32 届 ACM算法与体系结构并行性研讨会 (SPAA) 论文集，第 89 102 页，2020 – 。[7] Guy E. Blelloch、Jeremy T. Fineman 和 Julian Shun。贪心顺序极大独立集和匹配在平均情况下具有并行性。载于 ACM算法与体系结构并行性研讨会论文集，第 308 317 页，2012 – 。[8] Richard Cole。并行归并排序。《SIAM计算杂志》，17 785，1988 4 770( ): – 。[9] L. Dhulipala、G. E. Blelloch 和 J. Shun。使用压缩的纯函数树实现低延迟图流处理。载于 ACM编程语言设计与实现大会 (PLDI)，第 918–934 页，2019 年。[10] L. Dhulipala、C. Hong 和 J. Shun。Connectit：用于静态和增量并行图连通性算法的框架。《VLDB基金会会刊》 (PVLDB)，14(4):653–667，2020 年。[11] L. Dhulipala、C. McGuffey、H. Kang、Y. Gu、G. E. Blelloch、P. B. Gibbons 和 J. Shun。适用于非易失性内存的半非对称并行图算法。《VLDB基金会会刊》 (PVLDB)，13(9)，2020 年。[12] Laxman Dhulipala、Guy E. Blelloch 和 Julian Shun。理论上高效的并行图算法也能快速且可扩展。《ACM并行计算汇刊》，8(1):1–70，2021 年。[13] X. Dong、Y. Gu、Y. Sun 和 Y. Zhang。用于并行最短路径的高效步进算法及其实现。载于 ACM算法与体系结构并行性研讨会 (SPAA)，第 184–197 页，2021 年。[14] X. Dong、L. Wang、Y. Gu 和 Y. Sun。可证明快速且节省空间的并行双连通性算法。载于 ACM并行编程原理与实践研讨会 (PPOPP)，第 52–65 页，2023 年。[15] Xiaojun Dong、Yunshu Wu、Zhongqi Wang、Laxman Dhulipala、Yan Gu 和 Yihan Sun。用于半排序及相关问题的高性能、灵活并行算法。载于第 35 届 ACM算法与体系结构并行性研讨会 (SPAA) 论文集，第 341–353 页，2023 年。[16] Yan Gu、Julian Shun、Yihan Sun 和 Guy E. Blelloch。自顶向下并行半排序。载于 ACM算法与体系结构并行性研讨会论文集，第 24–34 页，2015 年。[17] Chase Hutton 和 Adam Melrod。并行批动态顶点着色，摊销更新时间为 $o(\log \delta)$，2025 年。[18] Joseph JáJá。《并行算法导论》。Addison-Wesley，1992 年。[19] Öjvind Johansson。图的简单分布式 $(\delta + 1)$ 着色。《信息处理快报》，70(5):229–232，1999 年。[20] Q. Liu、J. Shi、L. Dhulipala 和 J. Shun。并行批动态 k-core 分解及相关图问题。载于 ACM算法与体系结构并行性研讨会与体系结构 *(SPAA)*，2022 年。[21] Michael Luby。极大独立集问题的简单并行算法。载于*ACM*计算理论研讨会论文集 *(STOC)*，第 1–10 页，1985 年。[22] Michael Luby。在不损失处理器性能的情况下消除并行计算中的随机性。载于计算机与系统科学杂志，第 47 卷，286，1993 250pages – 。[23] Colin McDiarmid。论有界差分法。载于《组合学综述，*1989*》，伦敦数学会讲义系列第 141 卷，第 148–188 页。剑桥大学出版社，1989 年。[24] Mihai Pundefinedtraşcu 和 Mikkel Thorup。简单制表哈希的威力。《*ACM*期刊》，59(3)，2012 年 6 月。[25] J. Shi、L. Dhulipala 和 J. Shun。并行团计数和剥离算法。第 135–146 页。SIAM，2021 年。[26] J. Shi 和 J. Shun。蝴蝶计算的并行算法。载于*SIAM*计算机系统算法原理研讨会 *(APOCS)*，第 16–30 页。SIAM，2020 年。[27] J. Shun。实用并行超图算法。载于*ACM*并行编程原理与实践研讨会 *(PPOPP)*，第 232–249 页，2020 年。[28] T. Tseng、L. Dhulipala 和 G. Blelloch。批量并行欧拉游览树。载于第二十一届算法工程与实验研讨会 *(ALENEX)* 论文集，第 92–106 页。SIAM，2019 年。