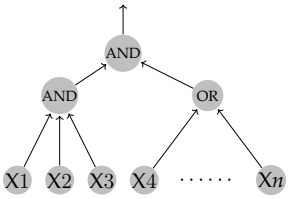
大数据算法

Big Data Algorithms

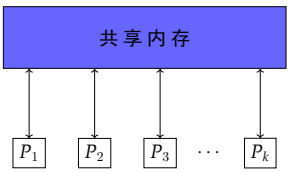
刘显敏
liuxianmin@hit.edu.cn

并行计算模型

- 分布式计算环境（1K、1M台机器）如何计算？



电路计算模型



PRAM计算模型

并行计算模型

- 电路计算模型：一个计算电路是一个有向无环图，包括
  - n个输入线路，标记为$X_1, \cdots, X_n$的布尔变量；
  - 逻辑与(AND)、或(OR)、非(NOT)门，根据输入计算输出结果；
  - 一个输出线路，返回布尔结果；
- 一个计算电路C本质上是一个布尔函数$C : \{0,1\}^n \mapsto \{0,1\}$
- 电路计算模型的计算资源：
  - 计算时间，电路的深度，即输入到输出的最长路径长度；
  - 处理器数目，电路中逻辑门的数量；
  - 扇入，逻辑门允许接收的最大输入数目；
- 复杂性类AC_i：在$O(\log^i n)$计算时间，无穷扇入，$n^{O(1)}$处理器数目时可解的问题；$AC = \bigcup_{i\in\mathbb{N}} AC_i$；
- 复杂性类NC_i：在$O(\log^i n)$计算时间，$O(1)$扇入，$n^{O(1)}$处理器数目时可解的问题；$NC = \bigcup_{i\in\mathbb{N}} NC_i$；
  - $AND(X_1, \ldots, X_n) \in AC_0$，$\notin NC_0$，$\in NC_1$；
  - $XOR(X_1, \ldots, X_n)$需要计算时间为$\Omega(\frac{\log n}{\log\log n})$的AC电路；

并行计算模型

- PRAM计算模型：包括
  - k个处理器；
  - 大小为M的共享内存；
- PRAM计算模型的计算资源：
  - 计算时间，模型求解问题所需的计算步数；
  - 计算量，$k\times$计算时间；

根据如何处理访问冲突，可以分为如下几种模型

- Concurrent Read Concurrent Write (CRCW) PRAM
  - 对同一单元的读写操作均可同时进行；
- Exclusive Read Exclusive Write (EREW) PRAM
  - 不允许对同一单元的同时读或者写操作；
- Concurrent Read Exclusive Write (CREW) PRAM
  - 可以同时读，但不允许同时写；
- Exclusive-read concurrent-write (ERCW) PRAM
  - 可以同时写，但不允许同时读；
- 即使CRCW模型，计算$XOR(X_1, \ldots, X_n)$至少需要$\Omega(\frac{\log n}{\log\log n})$的时间

并行计算模型

MapReduce模型

- 所有数据形如$\langle key, value\rangle$，计算分轮次进行
- 每一轮分为：Map、Shuffle、Reduce
- Map: 每个数据被map函数处理，输出一个新的数据集合
- Shuffle: 对编程人员透明，所有在Map中输出的数据，被按照key分组，具备相同key的数据被分配到相同的reducer
- Reduce: 输入$\langle k, v_1, v_2, \ldots\rangle$，输出新的数据集合

设计目标：更少的轮数、更少的内存、更少的工作量、更大的并行度

问题1：基本问题

构建倒排索引

- Map函数
  输入：⟨ docID, content⟩
  输出：⟨ word, docID⟩
- Reduce函数
  输入：⟨ word, (docID, ...)⟩
  输出：⟨ word, list of docID⟩

问题1：基本问题

单词计数

- Map函数
  输入：⟨ docID, content⟩
  输出：⟨ word, 1⟩
- Reduce函数
  输入：⟨ word, (1,1,1,...)⟩
  输出：⟨ word, count⟩

问题1：基本问题

检索Search

▷ Map函数
输入：⟨ (docID, lineno), content⟩
输出：⟨ docID, NULL⟩

▷ Reduce函数
输入：⟨ docID, (NULL, NULL, ...)⟩
输出：⟨ docID, NULL⟩

问题1：基本问题

矩阵乘法A

▷ Map()
- $\langle (A,i,j), a_{ij} \rangle \to \langle j, (A,i,a_{ij}) \rangle$
- $\langle (B,j,k), b_{jk} \rangle \to \langle j, (B,k,b_{jk}) \rangle$

▷ Reduce()
- $\langle j, (A,i,a_{ij}) \rangle, \langle j, (B,k,b_{jk}) \rangle \to \langle (i,k), a_{ij} * b_{jk} \rangle$

▷ Map+(): identity

▷ Reduce+()
- $\langle (i,k), (v_1, v_2, \dots) \rangle \to \langle (i,k), \sum v_i \rangle$

问题1：基本问题

矩阵乘法B

▷ Map()
- $\langle (A,i,j), a_{ij} \rangle \to \langle (i,x), (A,j,a_{ij}) \rangle$ for all x
- $\langle (B,j,k), b_{jk} \rangle \to \langle (y,k), (B,j,b_{jk}) \rangle$ for all y

▷ Reduce()
- $\langle (i,k), (A,j,a_{ij}) \rangle, \langle (i,k), (B,j,b_{jk}) \rangle \to \langle (i,k), \sum a_{ij} * b_{jk} \rangle$

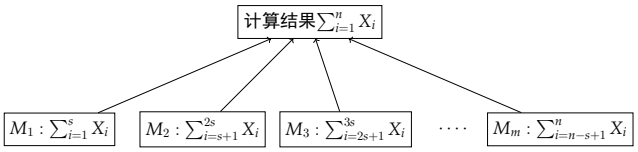
并行计算模型

▷ Massively Parallel Computation (MPC)计算模型
- m个处理器，处理N个数据，每台机器存储空间为s
  - ✓ 典型情况，$ms \geq N$，$ms = O(N)$，$s = N^{\epsilon}$，$m = O(N^{1-\epsilon})$
- 输入数据被分布式存储在所有机器上，通常考虑最坏情况
- 如果输出数据$\geq s$，也被分布式存储

▷ 计算时间按照轮次计算，每一轮包含如下操作
- 每个机器单独进行局部计算；
- 每个机器至多发送s个数据，至多接收s个数据；

问题2：求和问题

▷ 假设$s = \sqrt{n}$，计算$SUM(X_1, \dots, X_n)$



▷ $m = \sqrt{n} = s$，时间$O(1)$

▷ 当$s \ll \sqrt{n}$时，时间$O(\log_s n)$

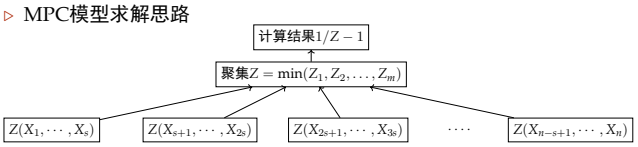
问题3：不重复元素数

▷ 假设$s = O(\sqrt{n})$，计算$DISTINCT(X_1, \dots, X_n)$

▷ FM+算法回顾，$O(\frac{1}{\epsilon^2})$代价，常数概率$1 \pm \epsilon$近似比

1. **for** j from 1 to k
2. 　随机选取一个哈希函数$h_j : [m] \mapsto [0,1]$，令$z_j = 1$;
3. 　每次遇到 i，更新: $z_j = \min(z_j, h_j(i))$;
4. $Z = \frac{1}{k} \sum_{j=1}^{k} z_j$;
5. **return** $1/Z - 1$.

▷ MPC模型求解思路



▷ 令$m = \sqrt{n}$，通信量$\leq m \times O(\frac{1}{\epsilon^2}) = O(\sqrt{n}) = s$，时间$O(1)$

▷ 当$s \ll \sqrt{n}$时，时间$O(\log_{\epsilon^2 s} n)$

问题4：排序问题

使用k台处理器，输入$\langle i, A[i] \rangle$

TeraSort: Round 1

map: $\langle i, A[i] \rangle$
1. 　输出 $\langle i\%p, ((i, A[i]), 0) \rangle$;
2. 　以概率r/n为所有$j \in [0, k-1]$输出 $\langle j, (A[i], 1) \rangle$;

reduce: $\langle j, (A[i], 1) \rangle$以及$\langle j, ((i, A[i]), 0) \rangle$
1. 　将$y = 1$的数据收集为S并排序;
2. 　构造$(s_1, s_2, \dots, s_{k-1})$，$s_l$为$S$中第$l\lceil \frac{|S|}{k} \rceil$个数据;
3. 　将$y = 0$的数据收集为D;
4. 　$\forall (i, A[i]) \in D$且$s_l < A[i] \leq s_{l+1}$输出$\langle l, (i, A[i]) \rangle$;

问题4：排序问题

使用k台处理器，输入$\langle i, A[i] \rangle$

TeraSort: Round 2

map: $\langle j, (i, A[i]) \rangle$
1. 　输出 $\langle j, (i, A[i]) \rangle$;

reduce: $\langle j, ((i, A[i]), \dots) \rangle$
1. 　将所有$(i, A[i])$收集并根据$A[i]$排序;

▷ 随机算法，$\mathbb{E}[|S|] = r = O(\frac{k \log(2n/\delta)}{\epsilon^2})$

▷ 样本点中的$k-1$个分点：如果s_l是S中的α分点，那么以很高概率，它处于$\alpha n \pm \epsilon n/k$

▷ 第二轮单机排序数据量$(1 \pm \epsilon)n/k$（概率至少$1 - \delta$）

概率基础

定理1.1[马尔可夫不等式]

对任意非负随机变量X和$a > 0$，有$\Pr(X \geq a) \leq \frac{\mathbf{E}[X]}{a}$

定义1.1[方差]

$$\mathbf{Var}[X] = \mathbf{E}[(X - \mathbf{E}[X])^2] = \mathbf{E}[X^2] - \mathbf{E}[X]^2$$

定理1.2[切比雪夫不等式]

对任意随机变量X和$a > 0$，有$\Pr(|X - \mathbf{E}[X]| \geq a) \leq \frac{\mathbf{Var}[X]}{a^2}$

定理1.3[切尔诺夫不等式，Chernoff/Hoeffding Bound]

令$X_1, X_2, \ldots, X_m$是独立的、取值$\in \{0, 1\}$的随机变量，变量和的期望为$\mu = \mathbf{E}[\Sigma_i X_i]$，令$\epsilon \in [0, 1]$，有$Pr[|\Sigma_i X_i - \mu| > \epsilon\mu] \leq 2e^{-\epsilon^2\mu/3}$.

内容总结

外存算法

亚线性空间算法　**大数据算法**　并行算法

亚线性时间算法

- ▷ 大数据计算课题组（海量数据计算研究中心）
- ▷ 联系方式：liuxianmin@hit.edu.cn