

1.1

设 p 个核的id分别是 $0..p-1$ ，给 0 至 $(n \bmod p)-1$ 号分配 $\lceil \frac{n}{p} \rceil$ 个， $n \bmod p$ 至 $p-1$ 号分配 $\lfloor \frac{n}{p} \rfloor$ 个

则

$$my_first_i = \begin{cases} id * \lceil \frac{n}{p} \rceil & id \leq n \bmod p \\ id * \lfloor \frac{n}{p} \rfloor + n \bmod p & id > n \bmod p \end{cases}$$
$$my_last_i = \begin{cases} (id + 1) * \lceil \frac{n}{p} \rceil & id < n \bmod p \\ (id + 1) * \lfloor \frac{n}{p} \rfloor + n \bmod p & id \geq n \bmod p \end{cases}$$

此公式对 $n \bmod p = 0$ 同样适用

1.6

- a) $p-1$
- b) $\lceil \log_2 p \rceil$

n	2	4	8	16	32
方法a	1	3	7	15	31
方法b	1	2	3	4	5

n	64	128	256	512	1024
方法a	63	127	255	511	1023
方法b	6	7	8	9	10

1.7

既是任务并行也是数据并行

对在树中深度相同的节点，可视为给每个核分配了一个子树执行树形累加，是数据并行

但每个核所做的任务本质上又是全局求和例子中的累加，和它一样是任务并行

1.9

GPU是训练神经网络的利器。神经网络主要基于简单的矩阵运算，但运算量巨大，一般的CPU难以胜任。GPU最初是用在图形化界面的渲染上，里面有大量能同时进行简单运算的处理单元，用以支持逐像素的渲染。近年来，由于发现了神经网络训练与图形界面渲染运算的相似性，GPU被用到了神经网络领域，使深度学习的算力极大提升，直接导致了近年来深度学习的井喷。在神经网络训练过程中，使用到了各式的并行计算技术。GPU的多个处理单元可认为在执行数据并行。大多数情况下，训练集很大，无法一次性读入到内存中，因此会采用GPU执行训练，CPU读取新的数据的方法，此时CPU与GPU可看做任务并行。我近期在做的一个CV相关的项目，所优化的指标只能使用

matlab在CPU上计算，对一张图片进行指标计算要花费近6分钟。为了更快的获得指标的值，我们选择了只对极少量的图片进行指标计算，且使用多台机器同时进行，这是十分典型的数据并行。在进行指标计算的同时，还在GPU上继续进行模型的迭代，这时CPU与GPU之间又是一个任务并行。