

GPU Computing



规约算法应用: 内积

Hui Liu

Email: hui.sc.liu@gmail.com

规约算法

- 求和
- 乘积
- 逻辑运算, 例如 and, or, xor...
- 极值, 例如 max, min.

$$V = \{v_0 = \begin{pmatrix} e_0^0 \\ \vdots \\ e_0^{m-1} \end{pmatrix}, v_1 = \begin{pmatrix} e_1^0 \\ \vdots \\ e_1^{m-1} \end{pmatrix}, \dots, v_{p-1} = \begin{pmatrix} e_{p-1}^0 \\ \vdots \\ e_{p-1}^{m-1} \end{pmatrix}\}$$

$$r = \begin{pmatrix} e_0^0 \oplus e_1^0 \oplus \dots \oplus e_{p-1}^0 \\ \vdots \\ e_0^{m-1} \oplus e_1^{m-1} \oplus \dots \oplus e_{p-1}^{m-1} \end{pmatrix} = \begin{pmatrix} \bigoplus_{i=0}^{p-1} e_i^0 \\ \vdots \\ \bigoplus_{i=0}^{p-1} e_i^{m-1} \end{pmatrix}$$

什么是内积

$$d = \langle x, y \rangle$$

Dot product of n -d vectors

$$v_1 = (a_1, a_2, a_3, \dots, a_n)$$

$$v_2 = (b_1, b_2, b_3, \dots, b_n)$$

$$v_1 \cdot v_2 = a_1 b_1 + a_2 b_2 + a_3 b_3 + \dots + a_n b_n$$

三阶段算法

- 1. 块大小, 256: 数组长度降低256倍
 - 大规模数组依旧很长, 例如 256万降低到1万
- 2. 对部分和继续使用上一步的算法
- 3. 使用一个块, 将最后结果规约

THANK YOU

