Analysis of parallel algorithms

p is number of processors
n is number of elements in array

1 Synchronous

Computation:
1. Compute squares in n operations
2. Sum them in n operations
3. Compute root in $O(1)$

$T_1 = 2n + O(1)$

2 Pairwise summation algorithm

Assuming $p \geq \frac{n}{2}$
Computation:
1. Compute squares in $O(1)$
2. Sum squares, which takes aproximately $\log_2 n$ operations, because on every step array is split into two equal halves
3. Compute root in $O(1)$

$T_p = \log_2 n + O(1)$

$S_p = \frac{2n}{log_2 n}$

$E_p = \frac{2n}{p \log_2 n} \leq \frac{2n}{\frac{n}{2} \log_2 n} = \frac{4}{\log_2 n}$

$\lim_{p \to \infty} E_p = 0$

3 Processor-aware summation

If $p \geq \frac{n}{2}$, this method is equivalent to previous.
So, let's assume $p < \frac{n}{2}$

Computation:
1. On every processor compute squares in section in $\frac{n}{p}$ operations
2. On every processor sum squares in $\frac{n}{p}$ operations
3. Sum results in ~ $\log_2 p$ operations (same logic as in previous algorithm)
4. Compute root in $O(1)$

$T_p = \frac{2n}{p} + \log_2 p$

$$S_p = \frac{2n}{\frac{2n}{p} + \log_2 p} = \frac{2}{\frac{2}{p} + \frac{\log_2 p}{n}}$$

$$E_p = \frac{1}{2 + \frac{p}{n} \log_2 p}$$