SPMD vector sum exact complexity

L. De Gaspari, M. Ronzani, A. Sanguineti

The SPMD vector sum problem

Summing up all the n elements present in a given vector using p parallel processors is done by allowing each processor to add two values in the first step and add another values to the sum of the previous ones at each step. This process results in a binary tree, with a layer for each time-step.
The purpose of this analysis is finding the exact time complexity of the algorithm.
Note that this analysis can be generalized to any vector reduction performed with a binary associative unit time operation.

Definition. Let $n \in [2, +\infty) \cap \mathbb{N}$ be the size of the input vector.

Definition. Let $p \in \mathbb{N}^+$ be the number of parallel processors dedicated to the problem on a PRAM.

In this case the time complexity with p processors, $T_p(n)$, is the maximum number of unit time operations taken by any of the p processor to complete its execution.

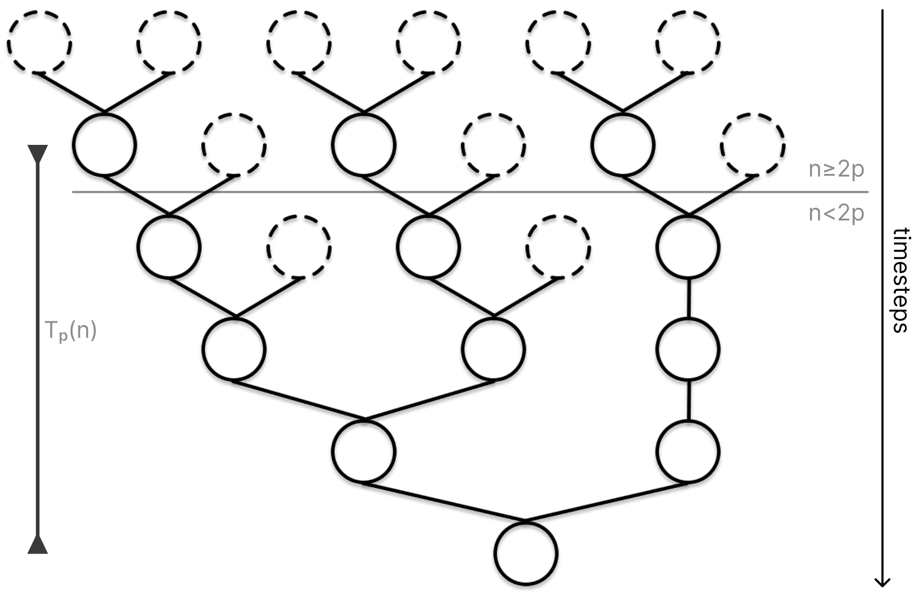

Figure 1: Tree example with $n = 11$ and $p = 3$, resulting in $T_p(n) = 5$.
Dashed circles are the read values, the other ones are computations.

The maximum number of sums that any processor performs in the above representation is the height of the left-most branch of the tree, coinciding with the number of time-steps.

observation 1. The exact time complexity, $T_p(n)$, coincides with maximum number of sums that any of the processors performs and with the height of the tree.

1 The recursion

From the definition of the problem we can derive the following recursion:

$$T_p(n) = \begin{cases} 1 + T_p\left(n-p\right) & n \geq 2p \\ 1 + T_p\left(\left\lceil \frac{n}{2} \right\rceil\right) & 2 < n < 2p \\ 1 & n = 2 \end{cases} \tag{1}$$

We will now proceed to simplify the two recursive cases.

2 The $n < 2p$ case

This is the case where we are left with enough processors to immediately reduce the leftover operations in a balanced binary tree, hence that tree can just be solved in:

Theorem. For any $n < 2p$, then:

$$T_p(n) = \lceil log_2 n \rceil \tag{2}$$

Proof. Proceeding by induction on $l = \lceil log_2 n \rceil$:

$$\textit{Base induction case} \ : \ \forall k \in (2^0, 2^1] \cap [2, 2p), \ T_p(k) = 1$$

However the only value of k that satisfies the above condition is 2.

$$\textit{By induction} \ : \ \forall k \in (2^{l-1}, 2^l] \cap [2, 2p), \ T_p(k) = l$$

For any $k \in (2^l, 2^{l+1}] \cap [2, 2p), \ T_p(k) = 1 + T_p\left(\left\lceil \frac{k}{2} \right\rceil\right)$, but $\left\lceil \frac{k}{2} \right\rceil \in (2^{l-1}, 2^l]$, so we can apply the induction hypothesis obtaining $T_p(k) = l + 1$.

□

3 The $n \geq 2p$ case

In this case we need to process p sums at every step until we reach a leftover number of addends less than twice the processors available, the non trivial part is figuring out how many steps are there before such threshold.

First we decompose $\frac{n}{p}$ in its integer part, denoted by $\lfloor\rfloor$, and fractional part, denoted by $\{\}$.

$$\frac{n}{p} = \left\lfloor \frac{n}{p} \right\rfloor + \left\{ \frac{n}{p} \right\}$$

$$n = p\left(\left\lfloor \frac{n}{p} \right\rfloor + \left\{ \frac{n}{p} \right\}\right) = p\left(\left\lfloor \frac{n}{p} \right\rfloor - 1\right) + p\left(\left\{ \frac{n}{p} \right\} + 1\right)$$

Lemma 1. Let $k, m \in \mathbb{N}$ so that $p \leq k < 2p$, then:

$$T_p(m + kp) = k + T_p(m) \tag{3}$$

Proof. $k \in [p, 2p) \Rightarrow k + p > 2p$

$$\begin{array}{rc} \textit{Base induction case} & T_p(m+p) = 1 + T_p(m) \\ \textit{By induction} & T_p(m+(k+1)p) = 1 + T_p(m+kp) = 1 + k + T_p(m) \end{array}$$

□

Theorem. By Lemma 1, with $k = \left\lfloor \frac{n}{p} \right\rfloor - 1 \in \mathbb{N}$ and $m = p\left(\left\{\frac{n}{p}\right\} + 1\right) \in [p, 2p) \cap \mathbb{N}$, we can state that:

$$T_p(n) = T_p\left(kp + m\right) = k + T_p\left(m\right)$$

$$T_p(n) = \left(\left\lfloor \frac{n}{p} \right\rfloor - 1\right) + T_p\left(p\left(\left\{ \frac{n}{p} \right\} + 1\right)\right) \tag{4}$$

Consequently by (2):

$$T_p(n) = \left(\left\lfloor \frac{n}{p} \right\rfloor - 1\right) + \left\lceil log_2\left(p\left(\left\{ \frac{n}{p} \right\} + 1\right)\right)\right\rceil \tag{5}$$

4 Meet $\widetilde{n}$ and $\widetilde{n}_{critical}$

Definition. Let $\widetilde{n} \in \mathbb{Z}_p$ so that $\widetilde{n} = p\left\{\frac{n}{p}\right\}$.

Then it follows that $n = \left\lfloor \frac{n}{p} \right\rfloor p + \widetilde{n}$. Consequently by definition:

$$\left\lceil log_2\left(p\left(\left\{ \frac{n}{p} \right\} + 1\right)\right)\right\rceil = \left\lceil log_2\left(p\left(\frac{\widetilde{n}}{p} + 1\right)\right)\right\rceil = \left\lceil log_2\left(\widetilde{n} + p\right)\right\rceil \tag{6}$$

observation 2. The following inequalities hold:

$$\lceil log_2 p \rceil \leq \lceil log_2(\widetilde{n} + p) \rceil \leq \lceil log_2 p \rceil + 1 \tag{7}$$

Proof. By definition $\widetilde{n} \in [0, p) \subseteq \mathbb{Q}$, since $\left\{\frac{n}{p}\right\} \in [0, 1) \subset \mathbb{Q}$, hence its true that:

$$p \leq \widetilde{n} + p < 2p$$

Since $log_2(\cdot)$ is a strictly increasing function we have:

$$log_2 p \leq log_2(\widetilde{n} + p) < log_2 p + 1$$

And by applying ceiling on all sides the observation is obtained, since ceiling is a non-decreasing function, and so $<$ becomes $\leq$. □

In the observation we re-obtained the same expression we got in (5) and (6) from $T_p(n)$, but this time we have two bounds for it. Furthermore since those bounds differ only by 1 and the bounded function is ceiled, an integer between two successive integers, that function is either exactly on the lower bound or on the upper bound.

$$\lceil log_2(\widetilde{n}+p)\rceil = \begin{cases} \lceil log_2 p\rceil & \widetilde{n}+p \leq 2^{\lceil log_2 p\rceil} \Leftrightarrow \widetilde{n} \leq 2^{\lceil log_2 p\rceil} - p \\ \lceil log_2 p\rceil + 1 & \widetilde{n}+p > 2^{\lceil log_2 p\rceil} \Leftrightarrow \widetilde{n} > 2^{\lceil log_2 p\rceil} - p \end{cases} \tag{8}$$

Definition. Let $\widetilde{n_c}$ "critical" as $\widetilde{n_c} = 2^{\lceil log_2 p\rceil} - p$.

So:

$$T_p(n) = \begin{cases} \left\lfloor \frac{n}{p} \right\rfloor - 1 + \lceil log_2 p\rceil & \widetilde{n_c} \geq \widetilde{n} \\ \left\lfloor \frac{n}{p} \right\rfloor + \lceil log_2 p\rceil & \widetilde{n_c} < \widetilde{n} \end{cases} \tag{9}$$

Theorem.

$$\left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil = \begin{cases} \left\lfloor \frac{n}{p} \right\rfloor & \widetilde{n} \leq \widetilde{n_c} \\ \left\lfloor \frac{n}{p} \right\rfloor + 1 & \widetilde{n} > \widetilde{n_c} \end{cases} \tag{10}$$

Proof. Remembering that $\widetilde{n} = n - \left\lfloor \frac{n}{p} \right\rfloor p$, we can derive the following.

$$\widetilde{n} \leq \widetilde{n_c} \Leftrightarrow n - \left\lfloor \frac{n}{p} \right\rfloor p \leq \widetilde{n_c}$$
$$\Leftrightarrow n - \widetilde{n_c} \leq \left\lfloor \frac{n}{p} \right\rfloor p \Leftrightarrow \frac{n-\widetilde{n_c}}{p} \leq \left\lfloor \frac{n}{p} \right\rfloor \Leftrightarrow \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil \leq \left\lfloor \frac{n}{p} \right\rfloor$$

The last "$\Leftrightarrow$" is justified by the monotonicity of ceiling, since we have a value on the left not higher than an integer on the right, his ceiling as well does not surpass such integer. Now by applying the ceiling function at both sides of the second-to-last inequality, the inequality is preserved thanks to ceiling being monotonic, and the last formulation is obtained.

$$\left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil \leq \left\lceil \left\lfloor \frac{n}{p} \right\rfloor \right\rceil = \left\lfloor \frac{n}{p} \right\rfloor \tag{11}$$

Similarly we can obtain another inequality.

$$n - \widetilde{n_c} > n - p \Leftrightarrow \frac{n-\widetilde{n_c}}{p} > \frac{n}{p} - 1$$
$$\Leftrightarrow \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil > \left\lceil \frac{n}{p} \right\rceil - 1 \geq \left\lfloor \frac{n}{p} \right\rfloor - 1$$
$$\Rightarrow \left\lfloor \frac{n}{p} \right\rfloor - 1 < \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil \Leftrightarrow \left\lfloor \frac{n}{p} \right\rfloor \leq \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil$$

From both inequalities arises the following:

$$\left\lfloor \frac{n}{p} \right\rfloor \leq \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil \leq \left\lfloor \frac{n}{p} \right\rfloor + 1 \tag{12}$$

Here we are still working with an integer between two successive integers, hence as before it is always exactly one of them, and the inequalities can be rewritten as the theorem. □

Finally the theorem can be substituted in $T_p(n)$, resulting in:

$$T_p(n) = \begin{cases} \left\lfloor \frac{n}{p} \right\rfloor - 1 + \lceil log_2 p \rceil = \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil - 1 + \lceil log_2 p \rceil & \widetilde{n_c} \geq \widetilde{n} \\ \left\lfloor \frac{n}{p} \right\rfloor + \lceil log_2 p \rceil = \left\lceil \frac{n-\widetilde{n_c}}{p} \right\rceil - 1 + \lceil log_2 p \rceil & \widetilde{n_c} < \widetilde{n} \end{cases} \tag{13}$$

In both cases we have the same expression for $T_p(n)$, hence we can unite the two.

$$T_p(n) = \lceil log_2 p \rceil + \left\lceil \frac{n - \widetilde{n_c}}{p} \right\rceil - 1 \tag{14}$$

And substituting $\widetilde{n_c}$:

$$T_p(n) = \lceil log_2 p \rceil + \left\lceil \frac{n - 2^{\lceil log_2 p \rceil}}{p} \right\rceil \tag{15}$$

5 Conclusion

Returning at once at the original expression for the recursion, we can substitute both cases' final expressions:

$$T_p(n) = \begin{cases} \lceil log_2 p \rceil + \left\lceil \frac{n-2^{\lceil log_2 p \rceil}}{p} \right\rceil & n \geq 2p \\ \lceil log_2 n \rceil & n < 2p \\ 1 & n = 2 \end{cases} \tag{16}$$