Abstract: In this paper, we tackle the problem of optimization of distributed video encoding on a single-level tree of processors with different computation and communication capacities. Specifically, we address the question - “what is the most efficient way to distribute the video load between the processors so that the total encoding time is the minimum”? We introduce the two-processor case as the basic building block of the single-level tree and derive the conditions under which optimum time performance can be obtained. Next, we extend the concept to a generic single-level tree with arbitrary number of processors. We show that by using the concept of processor equivalence, fast derivation of a feasible optimum load configuration is possible.

1 Introduction

One branch of scheduling theory that has gained wide importance over the last few years is - The Distributed Load Theory (DLT) [1]. The optimality principle of DLT states that the minimum execution time results when the all the processors involved in the computation complete their respective execution at the same time, as shown in figure 1. The two biggest shortcomings of DLT are that, (1) most calculations use homogeneous systems, and (2) the result-collection phase is ignored.

Figure 1: DLT optimality principle

In this paper, we tackle the problem of optimization of distributed video encoding on a heterogeneous single-level processor tree as shown in figure 2. The processors (nodes) have different computation and communication capacities. Further, we assume that the distributed processing is carried out in tree distinct, contiguous phases - load distribution, computation, and result collection. This means that if there are \( m \) child nodes, then \( (m!)^2 \) sets of \( m \) linear equations need to be solved to determine the optimum sequence. Instead, to enable fast calculation of a feasible solution, we construct the tree two processors at a time. We show that by the repeated application of the concept of equivalent processor, the entire tree can be collapsed into a representative processor and the distribution sequence can be determined.

Figure 2: Single-level heterogeneous tree

\[ p_0 \] is the root processor and \( p_1 \) and \( p_2 \) are the child processors. \( E_1, E_2 \) represent the computation time for the unit load on each processor, and \( C_1, C_2 \) represent the communication time for the unit load on their respective communication links with the root processor. \( \alpha_1, \alpha_2 \) are the fractions of the input load allocated to each processor such that \( \alpha_1 + \alpha_2 = 1. \) \( \delta \) is the compression ratio of the video encoder. The computation and communication times are assumed to be linear functions of the load fractions allocated to the processors. It can be proved that as long as \( C_2 > C_1, \) only the configurations I and II shown in figure 3 are important [2]. Further, it can be proved that as long as \( C_2 \) satisfies condition (1), configuration I is faster than configuration II [2].

\[
C_2 < C_1[1 + \theta(1 + \phi)] \quad \text{where} \quad \theta = \frac{C_2}{C_1 + E_1 + \delta C_1} \quad \text{and} \quad \phi = \frac{E_1}{C_2 + E_2 + \delta C_2} \tag{1}
\]
3 Equivalent Processor

As shown in figure 4, the two processors can be replaced by an equivalent processor with unit computation time $E_1^I$ connected to the root with an equivalent link with unit communication time $C_1^I$. The values of the parameters can be derived as [2]:

$$C_1^I = \frac{C_1C_2 + C_1E_2 + C_2E_1 + \delta C_1C_2}{E_1 + E_2 + \delta C_1 + C_2}$$  \hspace{1cm} (conf I) \hspace{1cm} (2)$$

$$C_1^I = \frac{C_1C_2 + C_1E_2 + C_2E_1 + \delta C_1C_2}{E_1 + E_2 + \delta C_1 + C_2}$$  \hspace{1cm} (conf II) \hspace{1cm} (3)$$

$$E_1^I = \frac{E_1E_2 - \delta C_1C_2}{E_1 + E_2 + \delta C_1 + C_2}$$  \hspace{1cm} (conf I) \hspace{1cm} (4)$$

$$E_1^I = \frac{E_1E_2}{E_1 + E_2 + \delta C_1 + C_2}$$  \hspace{1cm} (conf II) \hspace{1cm} (5)$$

4 Single-level Tree

By repeated application of the equivalent processor concept, two processors can be combined at a time either in configuration I or II after checking for condition (1), to generate an allocation tree as shown in figure 5. From the allocation tree, the timing diagram can be generated and hence the load fractions for the individual processors can be calculated. For example, in the special case where all processors satisfy condition (1), the load fractions are [2]:

$$\alpha_k = \frac{\prod_{j=k+1}^{m} f_j}{1 + \sum_{i=1}^{m-1} \prod_{j=i+1}^{m} f_j}$$, \hspace{1cm} \text{where} \hspace{1cm} (6)$$

$$f_{k+1} = \left( \frac{C_k + \delta C_k}{E_k + \delta C_k} \right)$$  \hspace{1cm} (7)$$

$$\alpha_m = \frac{1}{1 + \sum_{i=1}^{m-1} \prod_{j=i+1}^{m} f_j}$$  \hspace{1cm} (8)$$

5 Conclusion

In this paper, we showed that by considering the two-processor case as a basic building-block of the single-level processor tree and by using the concept of processor equivalence, it is possible to generate a feasible optimal solution for the load allocation in a short time.

In the future, we will revise our model to take into account various overheads and also to reflect other important practical considerations.