SPORT: A Near-Optimal Solution to Divisible Load Scheduling on Heterogeneous Systems

Abhay Ghatpande Hidenori Nakazato Hiroshi Watanabe

Graduate School of Global Information and Telecommunication Studies, Waseda University

1 Introduction

Divisible Load Theory (DLT) [1] is the mathematical framework to study Divisible Load Scheduling (DLS). But DLT ignores the result collection phase, and is also unable to deal with the general case where both the network links and computing nodes are heterogeneous.

In this paper, we first describe the DLS with result collection phase on a heterogeneous star network (DLSRCHET) problem in Sect. 2. In Sect. 3, we present a polynomial time algorithm, SPORT, (System Parameters based Optimized Result Transfer) as a near-optimal solution to the DLSRCHET problem. With the help of simulations, in Sect. 4, we show that the algorithm delivers consistent performance irrespective of the degree of heterogeneity of the underlying network and nodes. Finally, Sect. 5 provides the conclusion.

2 Problem Description

A divisible load (job) J is to be distributed to, and processed on a heterogeneous star network $\mathcal{N}$ as shown in Fig. 1. $\mathcal{N}$ consists of $m+1$ processors $p_0, \ldots, p_m$, and m links $l_1, \ldots, l_m$. For $k = 1, \ldots, m$, C_k is the inverse of the bandwidth of the link connecting node p_k to source p_0, and E_k is the inverse of the computation speed of p_k. The source p_0 splits J entirely into parts $\alpha_1, \ldots, \alpha_m$, and sends them to the respective processors $p_1, \ldots, p_m$ for computation, without retaining any part for itself. Each such set of m fractions is known as a *Load Distribution* α. Further, $0 < \alpha_k \leq 1$, and $\sum_{k=1}^{m} \alpha_k = 1$. This is the *normalization equation*.

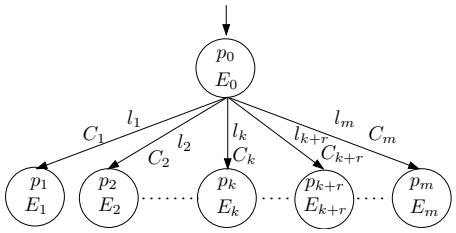

Fig. 1 Heterogeneous star network $\mathcal{N}$

It is assumed that the processors can communicate with only one other processor at a time, and cannot compute and communicate simultaneously. The execution of divisible job on each processor proceeds in three contiguous and distinct phases – distribution, computation, and result transfer. The time taken for communication and computation, as well as the volume of results generated is directly proportional to the load fraction α_k. The application-specific constant δ represents the ratio of size of output to input data. The time taken from the point when p_0 initiates communication with p_k, to the point when p_k completes the transfer of results back to p_0, is $T_k = \alpha_k C_k + \alpha_k E_k + \delta \alpha_k C_k$.

The DLSRCHET problem consists of finding the fraction of data to be allocated to each processor, the sequence of allocation of the data to the processors, and the sequence of result collection from the processors. Let σ_1 and σ_2 be permutations of order m such that $\sigma_1(i)$ and $\sigma_2(i)$ where $i = 1, \ldots, m$, respectively denote a mapping between the index of the allocation and collection sequence, and the processor number. Let x be the index in the collection sequence σ_2 of the processor to which p_0 distributes the load last in the allocation sequence. Then, $\sigma_1(m) = \sigma_2(x)$. We now formally state the DLSRCHET problem:

DLS WITH RESULT COLLECTION PHASE ON A HETEROGENEOUS STAR NETWORK (DLSRCHET)

Given a heterogeneous star network $\mathcal{N}$, and a divisible job J, find two permutations - σ_1, which determines the order of distribution of the load fractions from the source to the child processors, and σ_2, which determines the order of collection of computed results from the child processors, and a Load Distribution $\alpha = \{\alpha_1, \ldots, \alpha_m\}$, such that $0 < \alpha_k \leq 1$, and $\sum_{k=1}^{m} \alpha_k = 1$, so that the total execution time

$$T = \sum_{k=\sigma_1(1)}^{\sigma_1(m)} \alpha_k C_k + \alpha_{\sigma_1(m)} E_{\sigma_1(m)} + \sum_{k=\sigma_2(x)}^{\sigma_2(m)} \delta \alpha_k C_k$$

is minimized.

An exhaustive search of all possible permutations to find an answer to DLSRCHET has a complexity of order $O((m!)^2)$, which is impossible to solve in practice.

3 Proposed Algorithm

Algorithm 1 (SPORT) A heterogeneous star network $\mathcal{N}$ has $m+1$ processors $p_0, \ldots, p_m$, with p_0 as the source. Let the processors $p_1, \ldots, p_m$ be arranged such that $C_1 \leq C_2 \leq \ldots \leq C_m$. Define vectors $Dist[m]$ and $Coll[m]$ to store the distribution and collection sequences of the load fractions $\alpha_1, \ldots, \alpha_m$. Define variables C'_1, C'_2, E'_1, E'_2, α'_1, α'_2, j, k, $isSch\#1$, $isSch\#2$. Define the test condition

$$\frac{C'_1 C'_2 (E'_1 + E'_2 + C'_2 + \delta C'_2)}{(C'_1 + E'_1 + \delta C'_1)(C'_2 + E'_2 + \delta C'_2)} \geq (C'_2 - C'_1) \qquad (1)$$

and execute the following steps:

1. $Dist[\,] \leftarrow \{1, \ldots, m\}$, $Coll[1] \leftarrow Dist[1]$, $\alpha'_1 \leftarrow \alpha_1$ $C'_1 \leftarrow C_1$, $E'_1 \leftarrow E_1$, $isSch\#1 \leftarrow 0$, $isSch\#2 \leftarrow 0$
2. For k from 2 to m, do:
   (a) $C'_2 \leftarrow C_k$, $E'_2 \leftarrow E_k$, $\alpha'_2 \leftarrow \alpha_k$
   (b) If (1) == True, $isSch\#1 \leftarrow 1$, else, $isSch\#2 \leftarrow 1$.
   (c) If $isSch\#1 == 1$, do:
      i. $Coll[k] \leftarrow Dist[k]$
      ii. Save equation
      $$\alpha'_1 E'_1 + \delta\alpha'_1 C'_1 = \alpha'_2 C'_2 + \alpha'_2 E'_2$$
      iii. Assign
      $$\alpha'_1 \leftarrow \alpha'_1 + \alpha'_2$$
      $$C'_1 \leftarrow \frac{C'_1 C'_2 + C'_1 E'_2 + C'_2 E'_1 + \delta C'_1 C'_2}{E'_1 + E'_2 + \delta C'_1 + C'_2}$$
      $$E'_1 \leftarrow \frac{E'_1 E'_2 - \delta C'_1 C'_2}{E'_1 + E'_2 + \delta C'_1 + C'_2}$$
      iv. $isSch\#1 \leftarrow 0$, $k \leftarrow k + 1$, return to 2(a).
   (d) If $isSch\#2 == 1$, do:
      i. for j from 1 to k-1, do:
      $Coll[j+1] \leftarrow Coll[j]$, $Coll[1] \leftarrow Dist[k]$
      ii. Save equation
      $$\alpha'_1 E'_1 = \alpha'_2 C'_2 + \alpha'_2 E'_2 + \delta\alpha'_2 C'_2$$
      iii. Assign
      $$\alpha'_1 \leftarrow \alpha'_1 + \alpha'_2$$
      $$C'_1 \leftarrow \frac{C'_1 C'_2 + C'_1 E'_2 + C'_2 E'_1 + \delta C'_1 C'_2}{E'_1 + E'_2 + \delta C'_2 + C'_2}$$
      $$E'_1 \leftarrow \frac{E'_1 E'_2}{E'_1 + E'_2 + \delta C'_2 + C'_2}$$
      iv. $isSch\#2 \leftarrow 0$, $k \leftarrow k + 1$, return to 2(a).
3. Using the $m-1$ equations obtained above, and the normalization equation, form a set of m linear equations. Solve the equations to obtain load fractions $\alpha_1, \ldots, \alpha_m$ for the m processors. □

The logic behind the test condition, and details of values allocated to C'_1 and E'_1 can be found in [2]. Complexity of SPORT is of order $O(m^3)$.

4 Simulation Results

Simulations were carried out for four algorithms, viz., BruteForce, SPORT, FIFOC, and LIFOC. In BruteForce, the optimum distribution and collection sequences are found by evaluating all possible $(m!)^2$ sequences. Both FIFOC and LIFOC distribute load in the order of decreasing link bandwidth. In FIFOC, the result collection is in the same order as the distribution, while in LIFOC, it is in the reverse order as the distribution. In each simulation run, first the optimum time was found using BruteForce, and then the deviation of the execution time, ΔT, for the three variants SPORT, FIFOC, and LIFOC from the optimum was calculated. The values allocated to C_k and E_k in the different cases are approximately indicated in Fig. 2. The average ΔT over 1000 runs, $\langle \Delta T \rangle$, with $\delta = 0.1$ and $m = 5$ is plotted in Fig. 3. It can be

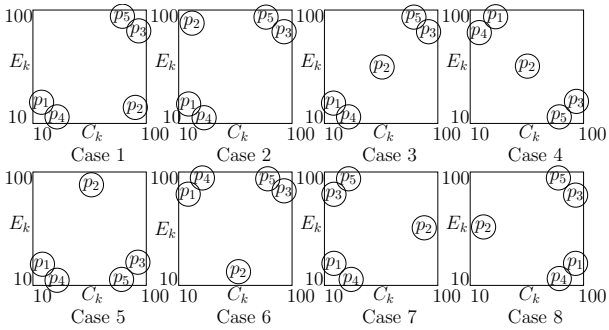

Fig. 2 Parameter selection for $m = 5$

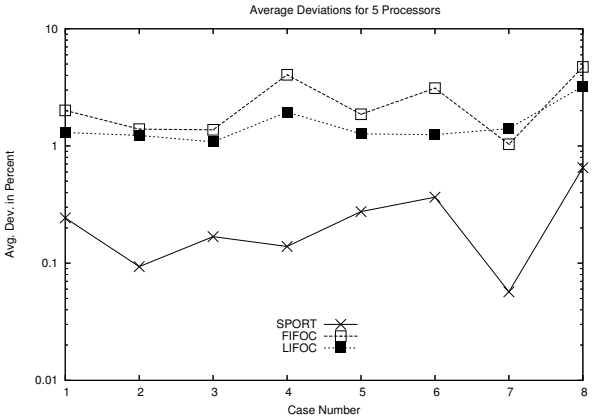

Fig. 3 $\langle \Delta T \rangle$ for $m = 5$

easily seen from Fig. 3, that SPORT performs exceedingly well as compared to FIFOC and LIFOC. On an average, FIFOC and LIFOC have errors of 881% and 537% of SPORT respectively. The maximum value of $\langle \Delta T \rangle$ of SPORT is 0.65% (case 8), which is still lower by 86% and 80% of FIFOC and LIFOC respectively.

5 Conclusion

We presented a polynomial time algorithm, SPORT, to solve the DLSRCHet problem, that predicts execution time with an error of less than 0.65% of the optimum. We found that SPORT is robust and simple, and consistently delivers near-optimal performance irrespective of the degree of heterogeneity of the system.

Bibliography

[1] V. Bharadwaj *et al.*: "Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems," Cluster Computing, vol. 6, no. 1, pp. 7–17, Jan. 2003

[2] A. Ghatpande *et al.*: "Divisible Load Scheduling with Result Collection Phase in a Heterogeneous Computing Environment," submitted for review to the IEICE Transactions on Communications