SPORT: Extended Simulations and Results for 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, after describing the DLSRCHET (DLS with Result Collection Phase on a Heterogeneous Star Network) problem in Sect. 2 and the SPORT (System Parameters based Optimized Result Transfer) algorithm in Sect. 3, we present the results of the simulations in Sect. 4. Section 5 gives the conclusion.

2 Problem Description

A divisible load J is to be 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 l_k 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 into parts $\alpha_1, \ldots, \alpha_m$, and sends them to the processors $p_1, \ldots, p_m$ for computation, without retaining any part for itself.

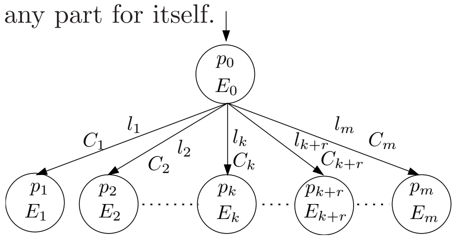


Fig. 1 Heterogeneous star network $\mathcal{N}$

Processors communicate with only one processor at a time, and cannot compute and communicate simultaneously. The execution of divisible load proceeds in three contiguous phases - distribution, computation, and result collection. The constant δ is the ratio of size of generated result data to allocated load. The time from the point when p_0 initiates communication with p_k, to the point when p_k completes the result transfer to p_0, is $T_k = \alpha_k C_k + \alpha_k E_k + \delta \alpha_k C_k$.

σ_a and σ_c are two permutations of order m, and each member $\sigma_a[k]$ and $\sigma_c[k]$, $k = 1, \ldots, m$, denotes the processor number that occurs at index k in the allocation and collection sequence respectively. $\sigma_a(k)$ and $\sigma_c(k)$, are *lookup functions* that return the index of processor $k = 1, \ldots, m$ in the allocation and collection sequence respectively. The DLSRCHET problem can be stated as:

Given a heterogeneous network $\mathcal{N}$ and a divisible load $\mathcal{J}$, find an allocation sequence permutation σ_a, and a collection sequence permutation σ_c, so that T is minimized, subject to the constraints:

$$T = \sum_{j=1}^{\sigma_a(k)} \alpha_{\sigma_a[j]} C_{\sigma_a[j]} + \alpha_k E_k + \sum_{j=\sigma_c(k)}^{m} \delta \alpha_{\sigma_c[j]} C_{\sigma_c[j]}, \quad k=1,\ldots,m \tag{1}$$

$$0 < \alpha_k < 1, \quad k = 1, \ldots, m \tag{2}$$

$$\sum_{k=1}^{m} \alpha_k = 1 \tag{3}$$

An exhaustive search of all permutations to find a solution to DLSRCHET has complexity of $O((m!)^2)$.

3 Proposed Algorithm

Algorithm 1 *In the heterogeneous star network $\mathcal{N}$, let the processors $p_1, \ldots, p_m$ be arranged such that $C_1 \leq C_2 \leq \ldots \leq C_m$. Define a test condition as:*

$$\frac{C_1 C_2 (E_1 + C_2 + E_2 + \delta C_2)}{(C_1 + E_1 + \delta C_1)(C_2 + E_2 + \delta C_2)} \geq (C_2 - C_1) \tag{4}$$

Execute the following:

1. $\sigma_a[\,] \leftarrow 1, \ldots, m$, $\sigma_c[1] \leftarrow \sigma_a[1]$, $C'_1 \leftarrow C_1, E'_1 \leftarrow E_1$
2. *For k from 2 to m, do:*
   (a) $C'_2 \leftarrow C_k$, $E'_2 \leftarrow E_k$
   (b) *Test (4) with $C_1 \equiv C'_1$, $E_1 \equiv E'_1$, $C_2 \equiv C'_2$, $E_2 \equiv E'_2$. If (4) true, goto 2(c), else, goto 2(d)*
   (c) i. $\sigma_c[k] \leftarrow \sigma_a[k]$
   ii. $$C'_1 \leftarrow \frac{C'_1 C'_2 + C'_1 E'_2 + C'_2 E'_1 + \delta C'_1 C'_2}{E'_1 + E'_2 + \delta C'_1 + C'_2}$$ $$E'_1 \leftarrow \frac{E'_1 E'_2 - \delta C'_1 C'_2}{E'_1 + E'_2 + \delta C'_1 + C'_2}$$
   iii. *$k \leftarrow k + 1$, and return to 2.a.*
   (d) i. *for j from 1 to $k-1$, do:* $\sigma_c[j+1] \leftarrow \sigma_c[j], \sigma_c[1] \leftarrow \sigma_a[k]$
   ii. $$C'_1 \leftarrow \frac{C'_1 C'_2 + C'_1 E'_2 + C'_2 E'_1 + \delta C'_1 C'_2}{E'_1 + E'_2 + \delta C'_2 + C'_2}$$ $$E'_1 \leftarrow \frac{E'_1 E'_2}{E'_1 + E'_2 + \delta C'_2 + C'_2}$$
   iii. *$k \leftarrow k + 1$, and return to 2.a.*
3. *Using σ_a and σ_c obtained above, form $m-1$ linear equations as follows: With $u = \sigma_a[\sigma_a(k) + 1]$,*

$$\forall k = 1,\ldots,m-1:$$

$$\alpha_k E_k = \begin{cases} \alpha_u C_u + \alpha_u E_u \\ \quad - \displaystyle\sum_{j=\sigma_c(k)}^{\sigma_c(u)-1} \delta \alpha_{\sigma_c[j]} C_{\sigma_c[j]} & \sigma_c(u) \geq \sigma_c(k)+1 \\ \alpha_u C_u + \alpha_u E_u \\ \quad + \displaystyle\sum_{j=\sigma_c(u)}^{\sigma_c(k)-1} \delta \alpha_{\sigma_c[j]} C_{\sigma_c[j]} & \sigma_c(u) \leq \sigma_c(k)-1 \end{cases}$$

4. *Along with (3), solve the complete set of m linear equations to obtain the respective load fractions $\alpha_1,\ldots,\alpha_m$ for the m processors.* □

SPORT is polynomial in m with complexity of $O(m^3)$.

4 Simulation Results

In the BRUTEFORCE algorithm, the optimum σ_a, σ_c are found by evaluating all $(m!)^2$ sequences. 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 of distribution.

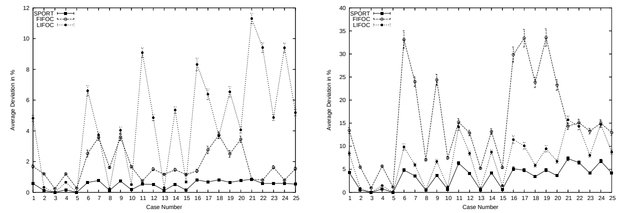
For $m = 4, 5, 6$, and $\delta = 0.2, 0.5, 0.8$, simulations were carried out for four algorithms: BRUTEFORCE, SPORT, FIFOC, and LIFOC. In each simulation run for the 25 cases in Table 1, the optimum time was found using BRUTEFORCE, and then the deviation of the execution time, ΔT, for SPORT, FIFOC, and LIFOC from the optimum was calculated. Mean deviation from optimal, $\langle \Delta T \rangle$, was calculated by averaging ΔT over 1000 simulation runs and plotted. All plots are not shown to conserve space.

Table 1 Parameters for Simulation

Case	$C_k \in$	$E_k \in$	Case	$C_k \in$	$E_k \in$
1	[1,10]	[1,10]	14	[10,100]	[1,100]
2	[1,10]	[10,100]	15	[10,100]	[10,1000]
3	[1,10]	[100,1000]	16	[10,1000]	[1,10]
4	[1,10]	[1,100]	17	[10,1000]	[10,100]
5	[1,10]	[10,1000]	18	[10,1000]	[100,1000]
6	[1,100]	[1,10]	19	[10,1000]	[1,100]
7	[1,100]	[10,100]	20	[10,1000]	[10,1000]
8	[1,100]	[100,1000]	21	[100,1000]	[1,10]
9	[1,100]	[1,100]	22	[100,1000]	[10,100]
10	[1,100]	[10,1000]	23	[100,1000]	[100,1000]
11	[10,100]	[1,10]	24	[100,1000]	[1,100]
12	[10,100]	[10,100]	25	[100,1000]	[10,1000]
13	[10,100]	[100,1000]			

From the plots in Figs. 2(a) and 2(b) for $m = 5$ and $\delta = 0.2, 0.8$, it is evident that SPORT has the lowest error, and consistently outperforms FIFOC and LIFOC in all the cases. For $\delta = 0.2$, LIFOC has a high error percentage, while for $\delta = 0.8$, the error percentage of FIFOC is increased. However, the error percentage of SPORT always remains below 7%.

Table 2 gives the maximum values of $\langle \Delta T \rangle$ for FIFOC, the case numbers when they occur, and the corresponding values of $\langle \Delta T \rangle$ for SPORT for those cases. Table 3 gives similar values for LIFOC. It is observed that SPORT outperforms FIFOC and LIFOC in all the cases. This proves that SPORT is able to generate good schedules for parameter values that cause FIFOC and LIFOC to perform poorly.



(a) $m = 5$, $\delta = 0.2$ (b) $m = 5$, $\delta = 0.8$

Fig. 2 Simulation Results

Table 2 max $\langle \Delta T \rangle$ of FIFOC

m	$\delta = 0.2$			$\delta = 0.5$			$\delta = 0.8$		
	FIFOC	case	SPORT	FIFOC	case	SPORT	FIFOC	case	SPORT
4	3.37	9	0.62	10.38	6	1.40	26.90	17	2.81
5	3.72	18	0.81	12.14	17	2.08	33.55	19	4.83
6	4.33	7	1.11	14.75	7	3.04	38.37	17	6.64

Table 3 max $\langle \Delta T \rangle$ of LIFOC

m	$\delta = 0.2$			$\delta = 0.5$			$\delta = 0.8$		
	LIFOC	case	SPORT	LIFOC	case	SPORT	LIFOC	case	SPORT
4	09.66	21	0.17	15.29	21	1.30	14.23	24	04.31
5	11.30	21	0.85	17.07	21	1.73	15.75	21	07.30
6	12.32	21	1.63	18.27	21	2.64	18.58	21	11.32

Table 4 max $\langle \Delta T \rangle$ of SPORT

m	$\delta = 0.2$	case	$\delta = 0.5$	case	$\delta = 0.8$	case
4	0.62	9	1.69	9	04.31	24
5	0.85	21	2.30	9	07.30	21
6	1.63	21	3.04	7	11.32	21

From Table 4, the maximum value of $\langle \Delta T \rangle$ of SPORT is 11.32%, and it occurs at case number 21 of $m = 6$, $\delta = 0.8$. It can be seen from Table 1 that in this case, the network links of the processors are much slower than their computation capacity.

5 Conclusion

We presented new simulation results and found that SPORT delivers near-optimal performance irrespective of the degree of heterogeneity of the system and value of δ. The maximum error of SPORT was 11.32% for all the configurations evaluated. To find the values of E_k and C_k that minimize the error is part of our future work. We will also work on multi-level trees.

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.*: "SPORT: A Near-Optimal Solution to Divisible Load Scheduling on Heterogeneous Systems," In Proc. IEICE Society Conference, pp. 7-8, Sep. 2005