Multi-Source Grid Scheduling for Divisible Loads

Thomas G. Robertazzi* and Dantong Yu**
*Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794, USA
**Department of Physics
Brookhaven National Laboratory
Upton, NY 11973, USA
tom@ece.sunysb.edu and dtyu@bnl.gov

***Abstract*— The applicability of min cost flow and multi-commodity flow mathematical programming problems to steady state, multi-source divisible load scheduling is examined. Applying the linear model concept of superposition to such steady state multi-source load distribution is suggested for linear and more general topologies. Finally, the use of heuristic optimization for a transient multi-source load distribution problem is discussed.**

I. Introduction

Over the past 17 years [1], [2] a good deal of research has been conducted on scheduling and load distribution with divisible loads. A divisible load is a data parallel load that can be arbitrarily partitioned among links and processors to gain the advantage of parallel processing. However most of this research has involved load distribution from a single source [3], [4]. That is, load originates from a single node in a larger grid or network. Multi-source load scheduling has received less attention but is a logical next step for research in this area. Not only can load be expected to originate from multiple sources in a grid, but even in a supercomputer like IBM Bluegene load is injected into the fabric of the machine from multiple points.

A 2002 paper on multi-source load distribution combining Markovian queueing theory and divisible load scheduling theory is Ko and Robertazzi [5]. In 2003 Wong, Yu, Veeravalli, and Robertazzi examined multiple source grid scheduling with capacity constraints [6]. Moges, Yu and Robertazzi considered multiple source scheduling for small size models via linear programming and closed form solutions in 2004 and 2005, respectively [7], [8]. Marchal, Yang, Casanova, and Robert in 2004 studied the use of linear programming to maximize throughput for large grids with multiple loads/sources [9].

This paper proposes the use of min cost flow and multi-commodity flow formulations for steady state divisible load scheduling with multiple sources (section 2). It then goes on in section 3 and 4 to discuss the use of superposition techniques, as used in electric circuit theory, for steady state divisible load scheduling in linear and more general topologies, respectively. Also in section 4 a heuristic approach for minimal time solution for transient divisible load models is discussed.

This paper is significant for proposing some new optimization approaches for the multiple source scheduling problem in grids. In particular, the use of superposition would simplify the conceptualization and computation of such steady state problems.

Dantong Yu's work is supported by DOE RHIC/ATLAS grants.

II. Flows as Mathematical Programming Problems

It has been known since Agrawal and Jagadish [2] that optimal load distribution and scheduling for divisible loads can be phrased as mathematical programming problems.

Here optimal load distribution when there are multiple sources of steady state load are formulated two ways: as a minimum cost flow problem and as a more general multi-commodity flow problem.

An advantage of using a minimum cost flow problem formulation is that costs can be assigned that are proportional to the amount of flow on each link. Here we assume that there is a single class of flow that is generated at sources and can be processed at any sink.

Let a flow on a link between adjacent nodes i and j be x_{ij} and the "cost" of the flow be c_{ij}. Then let the objective function for the minimum cost flow problem be

$$Z = \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

Here A is the set of links and the summation is over all links, also Z is the total cost.

Next, at each ith node one can write a mass balance constraint [10] that says that the difference between the flow out of node i (over all links leaving node i) and the flow into node j (overs all links to node j) is, thus

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b(i) \ for\ all\ i \in N \tag{2}$$

Here N is the set of nodes. If $b(i)$ is positive the ith node is a source and generates load. If $b(i)$ is negative, the ith node is a sink and processes load. Thus by setting the $b(i)$, the amount of load generated and processed at differences nodes can be include in the optimization. Note that steady state flows are being modeled here. Note also that often $\sum_{i\in N} b(i) = 0$.

With a final set of constraint equations one also optionally set lower (l_{ij}) and upper (u_{ij}) limits to the flow on the ijth link, that is

$$l_{ij} \le x_{ij} \le u_{ij} \ for\ all\ (i,j) \in A \tag{3}$$

Solution algorithms, for the minimum cost flow problem are simpler than for multi-commodity flow problems. However in phrasing the optimization as a multi-commodity flow problem on can process specific classes of load at specific processors. Le the kth load class be one of K classes. Then the objective function, which still retains the minimum cost flavor, is

$$Z = min \sum_{(i,j)\in A} \sum_{1\le k\le K} c_{ij}^k x_{ij}^k \tag{4}$$

Here c_{ij} and x_{ij} are the ijth link cost and flow, respectively, for the kth class,

A mass balance equation similar to (2) can be written for each Kth flow. Using the notation of Ahuja [10] one has

$$NX^k = b^k,\ k = 1, 2, \cdots, K \tag{5}$$

Thus depending on the sign of the b^k entires, there can be generation and processing of the kth class flow at each ith node. Optionally, upper limits can be placed on the amount of individual class flows and total flow across all classes on a link as the following two sets of constraint equations indicate:

$$0 \le x_{ij}^k \le u_{ij}^k \ for\ all\ (i,j) \in A\ and\ all\ k = 1, 2, \cdots, K \tag{6}$$

$$\sum_{1\le k\le K} x_{ij}^k \le u_{ij}\ for\ all\ (i,j) \in A \tag{7}$$

A final note is that an inequality can be included in the mass balance equation if the network/grid has more capacity for processing than the supply of load.

III. Linear Daisy Chains with Two Sources

In this and the following sections we examine a specific multi-source problem involving multiple sources with steady state load distribution.

It has been noted before that basic divisible load scheduling theory is a linear theory. Other linear theories such as electric circuit theory admit a principle of superposition. That is, the response of a network to multiple sources of excitation is equal to the sum of the responses to each source of excitation individually.

To date the theory of divisible loads has been quite successful in solving load distribution problems with a single source of load. Thus if superposition could be applied in the divisible load scheduling area, it would provide a useful means of computing the flow of load in different parts of a grid when there are multiple sources of load.

To see how this might work, consider a linear daisy chain of N processors with load originating in a steady sate sense (i.e. load/sec) from the two nodes at either end of the chain.

Consider the chain of Fig.(1) with flows from left most and rightmost nodes.

The left most node generates 24 units/second of load while the right most node generates 32 unit/second of load. Hypothetical load distribution patterns for both nodes are shown in the figure. If the two flows are superimposed algebraically the resulting superimposed flow at the bottom of the figure results.

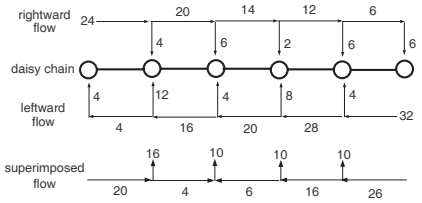

Fig. 1. Superimposed Flows in Linear Daisy Chain Network.

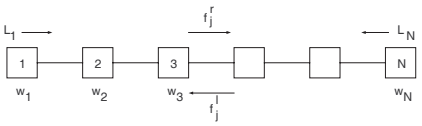

Fig. 2. Linear Daisy Chain Network.

Note that the superimposed flow has a minimum point at the third node from the left. It is straightforward to show that if the two individual flows are monotonically decreasing from source node through the geographic extent of the grid, then the magnitude of the superimposed flow will have a "bath tub" like function shape with a minimum somewhere on the grid. As a degenerate case if the rightward (leftward) flow is greater than the leftward (rightward) flow on all links a monotonically decreasing (increasing) flow magnitude from left to right results.

Let's try now to write some algebraic expressions for the flows in the linear daisy chain with two sources of load on the boundaries. Consider Fig.(2).

Load in the amount of L_1 units/second originates at the leftmost node while L_N units/second originate at the right most nodes. The rightward flow in the jth link (to the right of the jth node, $j = 1, 2, \cdots, N-1$) is f_j^r. The leftward flow is f_j^l. The inverse available processing speed of the jth node is w_j. We say "available" because in a grid a node may donate only part of its computational capability to process jobs.

We assume for this development that communication time is negligible compared to computation time. Thus it is as if one has links with very large communication speeds. In this case for a solution time optimal distribution of load, load should be distributed to each node in proportion to its computation speed. Over a finite time window load distribution will be balanced so that any other assignment will result in load being over assigned and under assigned to certain nodes and idle times developing. Following Wong [6], the optimal fraction of load to assign to each node is:

$$\alpha_i^{L_1} = \frac{\frac{1}{w_i}}{\sum_{k=1}^{N} \frac{1}{w_k}} \tag{8}$$

$$\alpha_i^{L_N} = \frac{\frac{1}{w_i}}{\sum_{k=1}^{N} \frac{1}{w_k}} \tag{9}$$

The flow f_j^r at link j is simply the generated load L_1 minus

the load consumed by nodes to the left of link j.

$$f_j^r = L_1 - \frac{\sum_{k=1}^{j} \frac{1}{w_k}}{\sum_{k=1}^{N} \frac{1}{w_k}} L_1 \quad (10)$$

$$= L_1 \left(\frac{\sum_{k=j+1}^{N} \frac{1}{w_k}}{\sum_{k=1}^{N} \frac{1}{w_k}}\right) \quad (11)$$

Similarly, for the leftward flow,

$$f_j^l = L_N \left(\frac{\sum_{k=1}^{j} \frac{1}{w_k}}{\sum_{k=1}^{N} \frac{1}{w_k}}\right) \quad (12)$$

The magnitude of the superimposed flow is

$$|f_j^r - f_j^l| = \frac{|L_1 \sum_{k=j+1}^{N} \frac{1}{w_k} - L_N \sum_{k=1}^{j} \frac{1}{w_k}|}{\sum_{k=1}^{N} \frac{1}{w_k}} \quad (13)$$

To find the minimum point of the flow magnitude, $|f_j^r - f_j^l| = 0$, so $L_1 \sum_{k=j+1}^{N} \frac{1}{w_k} = L_N \sum_{k=1}^{j} \frac{1}{w_k}$.

Therefore,

$$\frac{\sum_{k=1}^{j} \frac{1}{w_k}}{\sum_{k=j+1}^{N} \frac{1}{w_k}} = \frac{L_1}{L_N} \quad (14)$$

Thus the minimum point depends on the ratio of the two generated loads. For instance, if $L_1 = 2L_N$, the minimum point occurs where the amount of load consumed to the left of the minimum point is twice that consumed to the right of the point.

For a continuous and homogeneous version of the problem. (a continuum of nodes on a finite line from 0 to N) one has,

$$\frac{\int_0^x \frac{1}{w} dy}{\int_x^N \frac{1}{w} dy} = \frac{L_1}{L_N} \quad (15)$$

$$L_1 (N - x) \frac{1}{w} = L_N x \frac{1}{w} \quad (16)$$

$$x = \frac{L_1}{L_1 + L_N} N \quad (17)$$

This illustrates the same point about the minimum point location, x, just made above.

A natural question is to what extent is the superimposed flow solution optimal. Loads have been allocated to processors in proportion to computation speeds, a time optimal approach. Beyond this the superimposed flows in some sense minimize the amount of load that must be transported between nodes. The superposition technique illustrated here is similar to that an electric circuit with current sources and sinks.

IV. General Networks

The superposition technique of the previous section could be applied to more general grid networks that are two dimensional and that have multiple sources.

Consider the grid of Fig.(3(a)), the superimposed flow in $link_{ij}$ would be the algebraic sum of the flows due to each of the source individually.

In a very large grid it may make sense not to perform computation for a source at a too distant sink. That is, computation for a source might be done only on nodes in a local region close to the source. In fact for transient load distribution problems, it has been found for multilevel trees [3], a general spanning network, that for time optimal solutions much more load is processed close to the root (which is the load source) and very little in the further levels of a tree. Here communication delay is taken into account, unlike the situation discussed here.

We close by noting that this suggests an alternate load distribution optimization algorithm to the superposition based approach for transient problem. In this "heuristic" approach the grid is initially partitioned into local regions around each source as shown in Fig.(3(b)). The partition is evaluated by running a solution time optimal single source load scheduling algorithm, within each partition. A new partition could be created by transferring a node from one partition to another according to heuristic rules. Under a greedy strategy, the new partition is kept if it leads to an improved solution. More sophisticated heuristic approaches such as simulated annealing, tabu search or genetic algorithms could also be applied to this transient load distribution optimization problem. A partitioned linear daisy chain is considered by Lammie and Robertazzi in [11].

V. Conclusion

Several optimization techniques of the multiple source grid scheduling problem have been outlined. The application of superposition to such problems is particularly exciting as it brings an established linear modeling technique to a new area. In the process it provides conceptual simplification and the possibility of efficient computation.

Acknowledgment

The authors thank Professor Ester Arkin for very useful discussions.

References

[1] Y. Cheng and T. Robertazzi, "Distributed Computation with Communication Delays," IEEE Transactions on Aerospace and Electronic Systems, vol. 24, pp. 700–712, Nov. 1988.

[2] R. Agawal and H. V. Jagadish,"Partitioning Techniques for Large-Grained Parallelism," IEEE Transactions On Computers, vol. 37, pp. 1627–1634, Dec. 1988.

[3] B. Veeravalli, D. Ghose, V. Mani, and T. Robertazzi, "Scheduling Divisible Loads in Parallel and Distributed Systems," Los Alamitos, CA: IEEE Computer Society Press, Sept. 1996.

[4] B. Veeravalli, D. Ghose, and T. G. Robertazzi, "Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems," in special issue of Cluster Computing on Divisible Load Scheduling (T. G. Robertazzi and D. Ghose, eds.), vol. 6 of 1, pp. 7–18, Kluwer Academic Publishers, Jan. 2003.

[5] K. Ko and T. Robertazzi, "Scheduling in an Environment of Multiple Job Submissions, in Proceedings of the 2002 Conference on Information Sciences and Systems," (Princeton University, Princeton NJ), Mar. 2002.

[6] H. Wong, D. Yu, B. Veeravalli, and T. Robertazzi, "Data Intensive Grid Scheduling: Multiple Sources with Capacity Constraints," in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), (Marina del Rey, CA), Nov. 2003.

[7] M. Moges and T. Robertazzi, "Grid Scheduling Divisible Loads from Multiple Sources via Linear Programmin," in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2005), (Cambridge, MA), 2004.

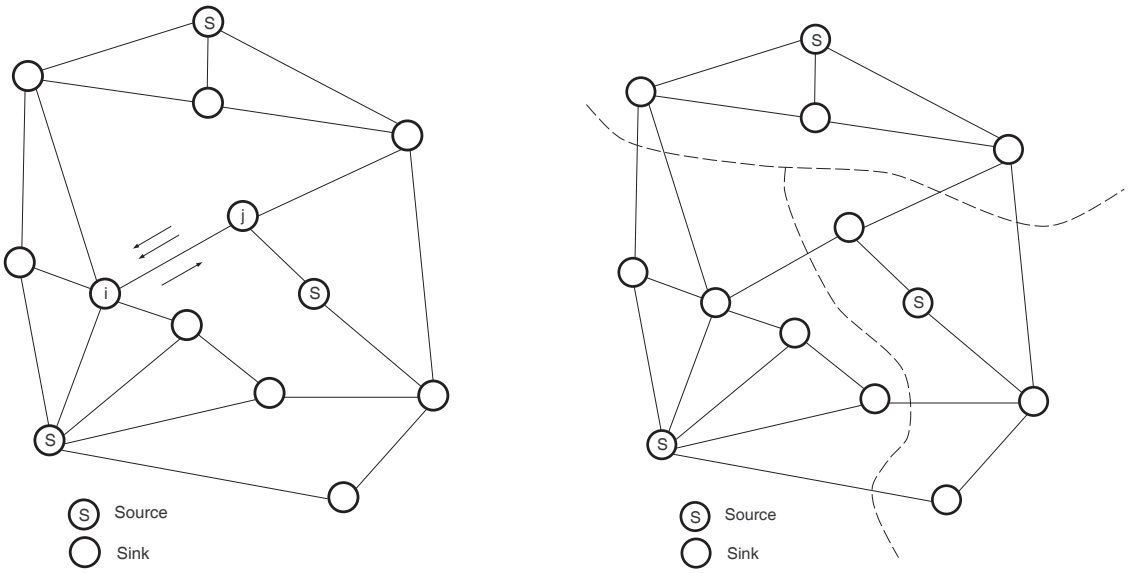

Fig. 3. a) Network of Processors b) Partition

[8] M. Moges, T. Robertazzi, and D. Yu, "Divisible Load Scheduling with Multiple Sources: Closed Form Solutions," in Proceedings of 2005 Conf. on Information Sciences and Systems, (The Johns Hopkins University, Baltimore, MD), 2005.
[9] L. Marchal, Y. Yang, H. Casanova, and Y. Robert, "A Realistic Network/Application Model for Scheduling Divisible Loads on Large-Scale Platforms," in International Parallel and Distributed Processing Symposium IPDPS'2005, IEEE Computer Society, Apr. 2005.
[10] R. Ahuja, T. Magnanti, and J. Orlin, eds., Network Flows. Prentice-Hall, 1993.
[11] T. Lammie and T. Robertazzi, "A Linear Daisy Chain with Two Divisible Load Sources," in Proceedings of 2005 Conf. on Information Sciences and Systems, (The Johns Hopkins University, Baltimore, MD), 2005.