SOME TASK ALLOCATION MODEL FOR DISTRIBUTED COMPUTING

Daniel C. Lee
Department of Electrical Engineering, University of Southern California
3740 McClintock Avenue, Los Angeles, CA 90089-2565
dclee@usc.edu

1. Introduction

A cluster (or distributed computing facilities), designed to perform a certain task, accomplishes its objectives by partitioning that task into subtasks, and by assigning subtasks to its agents. Generically, some of these subtasks interact; that is, they cannot be carried out by the corresponding agents in isolation. This introduces the need for communication between certain pairs of agents. In this paper, we focus on such communicational aspects of a cluster. In particular, we describe a cluster's organization by specifying "who talks to whom" or, mathematically, by means of an undirected graph $Go = (V_O, A_O)$, called the cluster's *organizational graph,* that specifies the communication capabilities available to the cluster. In particular, the nodes of G_O correspond to the agents, and the presence of an arc $(i,j) \in A_O$ signifies that agents i and j can communicate with each other. We will be always assuming that $(i,i) \in A_O$ for all $i \in V_O$, which expresses the natural fact that any agent can communicate with itself. Note that $(i,k) \notin A_O$ indicates that agent i and k cannot communicate, even if $(i, j) \notin A_O$ and $(j,k) \notin A_O$ for some i. A pair's inability to communicate with each other may model the cluster's security constraint. In some groups, in order to prevent the leakage of their secret information, the information is compartmentalized and kept separately by different agents who are prevented from communicating with each other. Or, such inability may model the case where maintaining reliable communication between the pair is prohibitively expensive.

Certain tasks might require communication between all agents of the cluster, in which case the most suitable cluster organization would correspond to a complete graph. On the other hand, there are numerous situations in which the task to be executed has a special structure, in which case fewer communication links suffice. Thus, the task assignment in the cluster is closely related to the organizational structure. For the case of fixed organizational structure (that is, the task assigner cannot control the which agent can communicate with which), the subtasks must be assigned in such a way that the organizational structure can accommodate the communication requirements. In this case, the major performance measure of the assignment may be the balance of loads among agents. For the case that the task assigner also has authority over the organizational structure, the task assignment is not constrained by the fixed organizational structure. However, each assignment requires a certain structure of inter-agent communications (i.e., organizational structure). The cost of keeping the communication structure may be an additional performance criterion in this case.

For our problem to be well defined, we need a mathematical representation of the communication requirements of the task to be executed. This is done in terms of another undirected graph $G_T = (V_T, A_T)$, called the *task graph.* The nodes of G_T correspond to subtasks, while the presence of an arc $(i, j) \in A_T$ signifies that subtasks i and j are interdependent. Each subtask $i \in V_T$ is to be assigned to an agent $\sigma_i \in V_O$, the agent primarily responsible for that task. In our model, the interdependence between two subtasks i and j is handled by assigning to a particular agent, denoted by $\sigma_{ij} \in V_O$, the responsibility of keeping track of this interdependence. It is then natural to require that σ_{ij} can communicate to both σ_i and σ_j .

Formally, we have the following definition. Given a task graph G_T, a *valid* organizational structure is defined as a graph G_O, together with a mapping σ: $V_T \cup A_T \rightarrow V_O$ such that $(\sigma_{ij},\sigma_i) \in A_O$ and $(\sigma_{ij},\sigma_j) \in A_O$ for every $(i,j) \in A_T$. (We will mostly use the notation σ_{ij} and σ_i instead of the more standard functional notations $\sigma(i)$ or $\sigma(i,j)$.)

The task allocation problems to be considered will be of the following form: given the task graph G_T, find the mapping σ and organizational structure G_O, subject to some additional constraints that remain to be specified so as to optimize a given performance measure. The following are some additional constraints:

a) We can impose a constraint on the cardinality of V_O, that is, on the number of available agents.
b) We could assume that the graph G_O is given, which would correspond to the case where we are dealing with a pre-existing cluster organization. In this case, all that remains to do is to design the mapping σ in some desirable way. An implicit assumption here is that all agents of the pre-existing organization are equally capable and versatile so that any subtask could be assigned to any agent
c) Going one step further, we could assume that the graph G_O is given and that the agent σ_i in charge of subtask i is also pre-specified for each i. In this case, we only have to choose which agent would be responsible for the handling of each subtask interaction. That is, we only need to choose the values of σ_{ij}, for every $(i,j) \in A_T$. Such a problem would correspond to a situation where each subtask is of a specific nature, intimately linked to a particular agent which is the only agent capable of handling it. On the other hand, the implicit assumption is that the handling of the interactions between subtasks i and j does not involve any particular expertise and can by handled by any agent, as long as the necessary communication links are in place.

Next, we have to specify some relevant performance criteria. Our first criterion pertains to load balancing. The agents of any cluster have limited resources and there is a limit on the number of their responsibilities. It is plausible that the agent assigned the largest number of responsibilities could be a bottleneck, and that its load should be minimized. Formally, we define the load ℓ_i of agent $i \in V_O$ to be the cardinality of the set $\sigma^{-1}(i)$. This is equal to the number of subtasks plus the number of interactions that this agent is responsible for. By defining the load this way, we are implicitly assuming that handling a subtask takes the same amount of resources with the handling of an interaction. The maximum load L is defined by $L = \max_{i \in V_O} \ell_i$.

Another performance criterion relates to the amount of communication resources employed by the cluster. This is a natural measure, given that communication is often a constrained resource. In fact, we will be considering two alternative ways of measuring communication resources.

A. Given a cluster organization G_O, let $C1$ be the number of arcs $(i, j) \in A_O$ for which $i \neq j$. Thus, $C1$ measures the number of communication links that have to be in place when setting up the cluster.
B. In an alternative method of measuring communication, we can measure the total amount of communication cost in the cluster. In particular, for every $(i, j) \in A_T$, agent σ_{ij} has to exchange messages with agents σ_i and σ_j, which leads, in general, to 2 units of communication traffic. However, if σ_{ij} coincides with σ_i, then we should not "charge" for communication between σ_{ij} and σ_i. Thus, the total communication traffic between all pairs of (distinct) agents, to be denoted by $C2$, can be defined as being equal to $2|A_T|$ minus the number of elements (i,j) of A_T for which $\sigma_{ij} \in \{\sigma_i, \sigma_j\}$.

It should be clear that the objectives of load balancing and low communication requirements compete with each other. For example, communication requirements are lowest if all subtasks are assigned to a single agent, resulting to a most unbalanced load. In our problem formulations, we will deal with this tradeoff by attempting to optimize one of the performance measures while constraining the other. For example, we might wish to minimize *C1* subject to a constraint that *L* be bounded by some given *L**.

Let us close by noting that the design problems that we have formulated are reminiscent of the mapping problems [Bo] that arise when subroutines are to be mapped to a parallel processing architecture. However, our problems have some distinctive features of their own, which make them different from the mapping problems that have been considered in the computer science literature. The formulation introduced in the present paper appears to be new.

The remainder of this paper is organized as follows. Each one of Sections 2,3,4 considers the problem under different assumptions on how much of G_O is assumed to be predetermined. For each choice of assumptions, we consider a few different problems depending on the particular choice of performance measure (*L, C1* or *C2).*

2. Fixed Cluster Organization

Let there be given a task graph G_T . In this section, we consider the cluster design problem under the assumption that the organizational graph G_o is also given, has the same number *n* of nodes as the task graph G_T, and we also have the constraint $\sigma_i = i$ for all *i*. Thus, it only remains to choose the value of σ_{ij} for every $(i,j) \in A_T$.

Note that it is easy to determine whether a valid cluster organization exists. In particular, we only need to check whether for every $(i, j) \in A_T$ there exists some *k* for which $(i, k) \in A_O$ and $(j, k) \in A_0$.

Minimizing the maximum load *L*
The first problem we consider is the following. We wish to find a valid task assignment which minimizes the maximum load *L*, subject to the constraints mentioned in the introduction to this section. This is equivalent to minimizing the maximum, over all agents *k*, of the number of pairs (i, j) $\in A_T$ assigned to that agent; equivalently, the number of pairs $(i, j) \in A_T$ for which $\sigma_{ij} = k$.

Theorem 2.1: *The above defined problem can be solved in polynomial time by solving a sequence of linear network flow problems.*

Minimizing a communication measure

The problem of minimizing the number *C*l of arcs is vacuous because G_O is assumed to be given and therefore *C1* is predetermined. The problem of minimizing *C*2 is also very simple, as we now discuss. If $(i, j) \in A_T$ and $(i, j) \in A_O$, then we should let σ_{ij} be equal to either σ_i or σ_j; if on the other hand, $(i, j) \notin A_O$, then we have to let σ_{ij} be equal to an arbitrary element *k* of V_0 such that $(i, k) \in A_O$ and $(j,k) \in A_t$. It should be clear that this method results in the minimal possible value of *C*2.

A more interesting problem is dealt with in the following result.

Theorem 2.2: *Consider the problem of minimizing C2 subject to an upper bound L* on the maximum load L. This problem can be formulated as a linear network flow problem and can be therefore solved in polynomial time.*

3. The Case where the Cluster Organization Is Given up to Isomorphism

Let there be given a task graph G_T. In this section we also assume that the graph G_O is given and has the same number of nodes as G_T. However, in contrast to the preceding section, we do not impose the requirement that $\sigma_i = i$ for all i. Instead, we impose the milder requirement that each division is assigned exactly one subtask, that is, the mapping $i \to \sigma_i$ is a permutation. Our main result states that even the problem of existence of a valid cluster organization is difficult.

Theorem 3.1: *The problem of deciding whether there exists a mapping σ such that the cluster (G_o, σ) is valid with respect to a given task graph G_T is NP-complete.*

Proof: That the problem belongs to NP is evident: if we have a YES instance, the mapping o provides a certificate.

We now note that the problem of interest is equivalent to the following:

***Problem P**:* Does there exist a permutation $i \to \sigma_i$ such that whenever $(i, j) \in A_T$, then the distance of σ_i and σ_j (in the graph G_O) is at most 2.

For any graph *G,* let *T(G)* be a graph with the same set of nodes and such that (i, j) is an arc of T(G) if and only if the distance of *i* and *j* in the graph *G* is at most two. We then see that we are dealing with the following problem:

***Problem P'**:* Given two graphs G_T and G_o with the same number of nodes, is G_T isomorphic to a subgraph of *T(Go)?*

We recall that the problem CLIQUE which is known to be NP-complete [GJ] and that CLIQUE problem is the following: Given a graph *G,* and an integer *k,* does G have a clique of size *k?*

Lemma 1: *CLIQUE remains NP-complete even if we restrict to instances for which $k \geq n/2 + 2$ and for which the degree of each node is at least $n/2 + 1$, where n is the number of nodes in the graph G.*

Recall now the SUBGRAPH ISOMORPHISM problem: given two graphs *G* and *G'*, is *G* isomorphic to a subgraph of *G'?* Since CLIQUE is a special case of SUBGRAPH ISOMORPHISM, and in view of Lemma 1, we see that SUBGRAPH ISOMORPHISM is NP-complete even if we restrict to graphs for which the degree of each node is at least $n/2 + 1$.

We will be needing another graph transformation. Given a graph *G,* we denote by *Q*(*G*) the graph which is the same as *G* except that each arc of *G* is replaced by a sequence of 3 arcs, as shown in Fig. 1. We introduce some more notation. If *G* is a graph and *i* is a node of that graph, we use *T*(Q(*i*)) to denote the image of node *i* when the transformations *Q* and *T* are applied in succession.

Lemma 2: *Let G be a graph in which all nodes have degree at least d.*

a)

If i is a node of G, then T(Q(i)) has degree at least 2d; all nodes of T(Q(G)), not of the form T(Q(i)) for some i, have degree bounded by n + 1.

b)

If (i, j) is an arc of G, then the distance, which is in the graph T(Q(G)), between T(Q(i)) and T(C)(j)) is equal to 2; if (i, j) is not an arc of G, then the distance between T(Q(i)) and T(Q(j)) is larger than 2.

Note that if all nodes of *G* have degree at least $n/2 + 1$, then nodes, of the form *T(Q(i))* will have degree at least $n + 2$. All other nodes of *T(Q(G))* will have degree at most $n + 1$. Thus, for each node of *T(Q(G)),* it can be immediately determined whether it is of the form *T(Q(i)) or* not.

Lemma 3: *Let G and G' be graphs in which all nodes have degree at least n/2 + 1. Then, G is isomorphic to a subgraph of G' if and only if T(Q(G)) is isomorphic to a subgraph of T(Q(G')).*

We notice that Lemma 3 reduces a special case of SUBGRAPH ISOMORPHISM (shown earlier to be NP-complete) to ***problem P'***, with the identification $G_o = Q(G')$ and $G_T = T(Q(G))$, except that we have not enforced the requirement that the two graphs in an instance of ***problem P'*** have the same number of nodes. This is easily taken care of, by adding a number of zero-degree nodes, and we conclude that ***problem P'*** is NP-complete, and the proof of Theorem 3.1 has been completed. **Q.E.D.**

We have shown that it is difficult to even determine whether a valid cluster does exist. It follows that the problem of determining an optimal valid cluster is also difficult (NP-hard), for any nontrivial choice of the performance criterion.

4. The Case where Only the Number of Nodes in *Go* Is Fixed

We now consider the case where G_T is given and we require that G_O have the same number of nodes as G_T; no other constraints are imposed on G_O. We also impose the requirement that each node *i* of G_T be mapped to a different node σ_i of G_T.

Under the above constraints, the problem of designing a valid organization that minimizes C1 is trivial: assuming that G_T is connected with *n* nodes, let $G_O = (\{1,...,n\}, \{(1,2),...,(1,n)\})$ (a "star" graph), and let $\sigma_{ij} = 1$ for all $(i,j) \in A_T$. We then have C1=n-1. Since G_T is connected, it is clear that G_0 must also be connected and therefore no valid organization could have less than n-1 arcs.

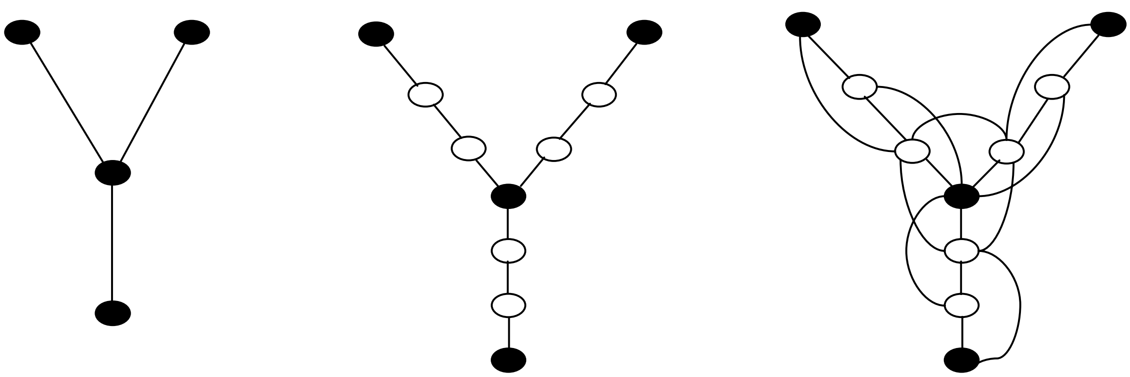
If we impose a load balancing constraint $L \leq L^*$ and attempt to minimize *C*l subject to that constraint, we obtain an apparently more difficult problem. We conjecture that this problem is NP--hard [GJ], although we have not been able to establish this result.

The last problem to be considered is dealt with by the following result.

Theorem 4.1: *Under the assumptions of this section, the problem of designing a valid organization in which C2 is minimized subject to the constraint $L \leq L^*$, can be formulated as a min-cost linear network flow problem and can be solved in polynomial time.*

5. Conclusions

We have formulated a new class of design problems for clusters. We have derived solution procedures for some of these design problems, and we have showed that another variation leads to NP-hard problems. We believe that our formulation captures some generic features of cluster design problems.



a) **b)** **c)**

Figure 1: a) A graph *G*; b) the graph *Q(G)*; c) the graph *T(Q(G))*.

References

[GJ] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company: New York, 1979.

[Bo] Bokhari, S.H., "On the mapping problem", *IEEE Tran. on Computers, Vol. 30,* pp. 550-557, 1981.

[B87] Bokhari, S.H., *Assignment Problems in Parallel and Distributed Computing*, Kluwer Academic Publishers, Boston: MA, 1987.