Efficient Granularity and Clustering of the Directed Acyclic Graphs

Qiangsheng Hua[1],
Student Member, ACM, IEEE and IEEE Computer Society and Communication Society
Zhigang Chen[2]
College of Information Science&Engineering, Central South University, Changsha (410083),China

[3]*Abstract:* The granularity of the DAG which determines the trade-off between communication and execution costs is the most important factor in DAG scheduling. So this paper firstly gives a DAG model and its clustering problem; Secondly, it addresses the drawbacks of the existing granularity definitions; Thirdly, it proposes a new granularity definition of the DAG which is very useful for the large communication delay DAGs. Based on this new definition, the paper proves that nonlinearly clustering is better for linearly clustering for the fine grain DAGs; Finally, it gives the bounds for linearly clustering for the DAGs.

Keywords: DAG, Granularity, Linearly Clustering, Nonlinearly Clustering, performance bound

I. Introduction

In the last three decades, many computation models in scheduling field have been studied. From the Directed Graph[1] which doesn't take the communication costs into account to the node-labeled and edge-labeled Directed Acyclic Graph[2],both of them have attracted many researchers' attention. But nowadays, more and more experts have realized that the communication costs significantly affect the performance of the parallel systems, so the latter model prevails over the first one during the last two decades.

II. DAG model and clustering problem

A weighted DAG is a tuple $G=(V,E,T,C)$,where $V=(V_1,V_2,...,V_{|V|})$ is the set of nodes and $|V|$ is the number of the nodes, $E=\{e_{ij}|v_i,v_j \in V\} \subseteq V \times V$, is the set of communication edges and $|E|$ is the number of the edges. The set C is the set of edge communication costs and T is the set of node computation costs. The value $C_{ij} \in C$ is the communication cost incurred along the edge $e_{ij} \in E$, which is zero if both nodes are mapped in the same processor. The value $T_i \in T$ is the computation cost for node $V_i \in V$.

Now we will study the problem of clustering directed acyclic graphs(DAGs). Clustering is a mapping of the nodes of a DAG onto m clusters. Every task in a cluster must execute in the same processor. A clustering is called linear if every cluster is one simple directed path in the task graph; otherwise is called nonlinear. Clustering has been used as a pre-processing step in scheduling task graphs on parallel architectures[3].

This work was supported by the Hunan Natural Science Foundation of China under the grant No.02JJY2097. Email: qshua@computer.org, czg@csu.edu.cn

III. Analyses of the existing granularity definitions

Here the grain definition refers to the grain of the tasks, and the granularity definition refers to the granularity of the Directed Acyclic Graph. In paper[2], a grain is defined as one or more concurrently executing program modules. A grain begins executing as soon as all of its inputs are available, and terminates only after all of its outputs have been computed. The grain size is defined as the number of atomic operations in the grain. Grain packing problem means reducing total execution time by balancing execution time and communication time. In paper[4][5],a grain is defined as a set of program steps or instructions that is to be executed sequentially by a single processor. These steps will necessarily be executed sequentially. In paper[6],a grain is even defined as the execution time between synchronizations. Because the above definitions of the grain just refers to the individual task of the DAG and they only take the computation costs into account without the communication costs, they are definitely not the powerful tool in the DAG scheduling field. Fortunately, the granularity of the DAG including both the computation and communication comes into being in paper[3,9-12].

Paper[7] defines the grain of the task as the ratio R/C(R stands for the computing units of time and C stands for the communication units of time) and shows that it is this ratio that determines the optimum trade-off point between parallelism and sequentialization. Paper[8] has extended [7]'s grain definition for arbitrary DAG's and have introduced a new quantity called the granularity of the DAG, defined as $g=\min_{x=1..v}\{t_x / \max_j c_{x,j}\}$

where c_{xj} are communication costs of the edges going out from node n_x and t_x is the computation cost of that node, v stands for the number of the nodes. Another granularity presents in paper[9] defines the granularity of a DAG as the average ratio of the node weight to maximum adjacent outgoing edge weight. Finally, paper[3] extends paper [8]'s granularity definition and introduced the fork/join granularity of the DAG. It is briefly introduced below:

A DAG consists of fork or/and join sets such as the ones shown in figure1. The join set J_x consists of all immediate predecessors of node n_x. The fork set F_x consists of all immediate successors of node n_x. Let $J_x=\{n_1,n_2,...,n_m\}$ and define

$g(J_x)=\min_{k=1..m}\{t_k\} / \max_{k=1..m}\{c_{k,x}\}$

Similarly let $F_x=\{n_1,n_2,\ldots,n_m\}$ and define

$$g(F_x)=\min_{k=1..m}\{t_k\}/\max_{k=1..m}\{c_{x,k}\}$$

Then it introduces the grain of a task as

$g_x=\min\{ g(J_x), g(F_x)\}$

and the granularity of a DAG as

$$g(G)=\min_{x=1..v}\{g_x\}$$

It calls a DAG coarse grain if $g(G)\geq 1$.

This kind of granularity definition proves to be very useful for the small communication delay(coarse grain) DAGs. And there are some promising theorems for this definition:

1. For any nonlinear clustering of a coarse grain DAG, there exists a linear clustering with less or equal parallel time, and
2. For any linear clustering algorithm,

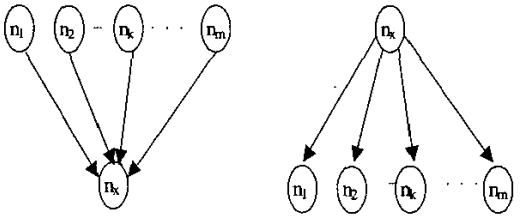
$$PT_{opt}\leq PT_{lc}\leq(1+1/g(G))PT_{opt}$$

where PT_{opt} is the optimum parallel time and PT_{lc} is the parallel time of the linear clustering. And moreover for a coarse grain DAG:

$$PT_{lc}\leq 2*PT_{opt}$$

A detailed discussion of this granularity can be found in paper[3].

From the above mentioned, we can see that the granularity in paper[3] is more promising and reasonable. So many subsequent researches in this field also refer to



(a) Join set J_x (b) Fork set F_x

Figure 1: Fork and join sets

this granularity[10-12] definition. But we can clearly notice that this definition is only very useful for the coarse grain(small communication delay) DAGs. It can't play an important role in the large communication delay DAGs. So we have our another new definition of the DAG granularity.

IV. A New Granularity Definition

As a matter of fact, our new definition is similar to Yang's definition[3], or we can call it his counterpart. Because Yang's granularity pays more attention to the small communication delay DAG(It is a coarse grain DAG if $g(G)\geq 1$.),our definition pays more attention to the large communication delay DAG, and it is also a fork/join granularity definition. The fork and join sets are shown in figure1.

We now define

$$G(J_x)=\max_{k=1..m}\{t_k\}/\min_{k=1..m}\{c_{k,x}\} \quad \text{and}$$

$$G(F_x)=\max_{k=1..m}\{t_k\}/\min_{k=1..m}\{c_{x,k}\}$$

Then we introduce the grain of a task as

$G_x=\max\{G(J_x), G(F_x)\}$

and the granularity of a DAG as

$$G(G)=\max_{x=1..v}\{G_x\}$$

We call a DAG fine grain if $G(G)\leq 1$.

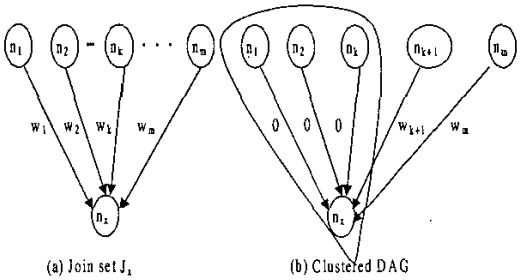
Now we will show that nonlinearly clustering is better than the linearly clustering for the fine grain DAGs. This can be proved via the join set graph.(fork set is the same with the join set).

We consider the join set J_x with $c_{j,x}=w_j$, j=1..m, see figure2(a), and without loss of generality, we assume that the nodes and edges are sorted such that $t_j+w_j\geq t_{j+1}+w_{j+1}$, j=1..m-1,where t_j is the computation cost of that node. Then the optimal schedule length is equal to:

$$\max\{t_x+\sum_{j=1}^{k}t_j\ ,\ t_x+w_{k+1}+t_{k+1}\}$$

where k is given by the following conditions:

(1) $\sum_{j=1}^{k}t_j\leq w_k+t_k$ and

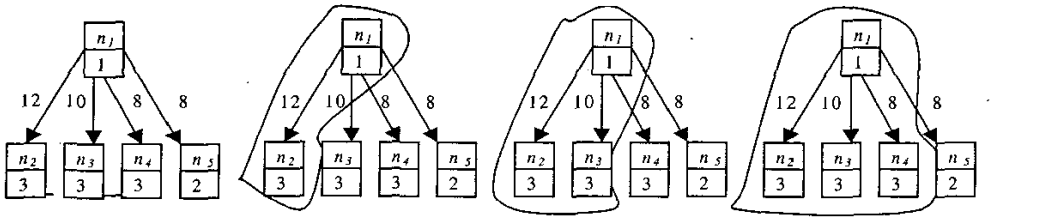


(a) Join set J_x (b) Clustered DAG

Figure 2: An optimum clustering algorithm for the join set

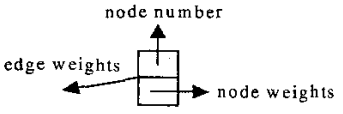
2) $\sum_{j=1}^{k+1}t_j>w_{k+1}+t_{k+1}$

Here note that the edge cost in the same cluster is zero. From the 1st inequality, paper[3] analyzes that for a coarse grain DAG, linearly clustering is better for the nonlinearly clustering. And now from the 2nd inequality we will prove that the nonlinearly clustering is better than the linearly clustering and the finer the granularity of the DAG, the more the edges will be clustered in the same cluster. From the 2nd inequality, we can get



(a)original DAG (b)linearly clustering (c)1st nonlinearly clustering (d)2nd nonlinearly clustering

Figure 3: A fine grain DAG and its three different clustering



$$\sum_{j=1}^{k} t_j > w_{k+1} \quad \text{and}$$

$$k^* \max_{j=1..m}\{t_j\} \geq \sum_{j=1}^{k} t_j > w_{k+1} \geq \min_{j=1..m}\{w_j\}$$

It then is reduced to:

$$k \geq 1/\{\max_{j=1..m}\{t_j\}/\min_{j=1..m}\{w_j\}\} = 1/G(G)$$

Now we can concluded that if G(G)<1 then this inequality implies that $k_{opt} \geq 2$ (since k is an integer), which means that the optimum clustering is a nonlinearly clustering. And the smaller the G(G), the larger number of the k_{opt}.

V. Bounds for linearly clustering for DAGs

From the two definitions of G(G) and g(G),we can reach the following conclusions:

$G(G) \geq g(G)$;

The following theorem 1 is true for previous two popular granularity definitions(paper[3][8]) and our new definition.

Theorem 1:

For any linearly clustering algorithm, we have

$$PT_{opt} \leq PT_{lc} \leq \{(1+G(G))/g(G)\} * PT_{opt}$$

Where PT_{opt} is the optimum parallel time and PT_{lc} is the parallel time of the linear clustering.

Proof:

The proof is similar to paper[3].Let L_{cp} be the length of a critical path $cp=\{n_1,n_2,\ldots,n_k\}$of G which includes edge communication costs. We have

$$L_{cp} = \sum_{j=1}^{k} t_j + \sum_{j=1}^{k-1} c_{j,j+1}$$

And from the definition of G(G) and g(G),we have

$$\sum_{j=1}^{k-1} c_{j,i+1} \leq \sum_{j=1}^{k} t_j / g(G)$$

and $$\sum_{j=1}^{k} t_j \leq \sum_{j=1}^{k-1} c_{j,i+1} * G(G)$$

Consider the final clustered DAG G_m which is the same as G with the exception that the edge weights of the clusters in G have been set equal to zero. Then PT_{lc} is equal to the length of the critical path of G_m. We conclude that

$$\sum_{j=1}^{k} t_j \leq PT_{opt} \leq PT_{lc} \leq L_{cp}$$

Then

$$PT_{lc} \leq L_{cp} = \sum_{j=1}^{k} t_j + \sum_{j=1}^{k-1} c_{j,i+1} \leq (1+G(G)) *$$

$$\sum_{j=1}^{k-1} c_{j,i+1} \leq \sum_{j=1}^{k} t_j \{(1+G(G)/g(G)\}$$

$$\leq \{(1+G(G)/g(G)\} * PT_{opt}$$

Here when G(G)=1, then this inequalities are reduced to

$$PT_{opt} \leq PT_{lc} \leq \{1+1/g(G)\} * PT_{opt}$$

When g(G)=1, then this inequalities are reduced to

$$PT_{opt} \leq PT_{lc} \leq \{1+G(G)\} * PT_{opt}$$

And when G(G)=g(G), then this inequalities are reduced to

$$PT_{opt} \leq PT_{lc} \leq 2\ PT_{opt}$$

Some of the results are the same with previous papers got[3].

VI. An experimental result

A fine grain DAG and its three different clustering graph is shown in figure 3. From our granularity definition, we can see that the granularity of the original DAG is

$$G(G)=3/8$$

so it is a fine grain DAG, the optimal scheduling length must be based on a nonlinearly clustering According to the following formula:

$$k \geq 1/G(G)=8/3$$

we know that the $k_{opt}=3$. So the optimal clustering is figure3(d). The following analyses will explain why the figure3(d) is optimal.

Figure3(b) is a linearly clustering, the scheduling length is 14. In figure3(c), there are two different directed paths in the same cluster, so it is a nonlinearly clustering, but the k=2. The scheduling length of this graph is 12. Figure3(d) is another nonlinearly clustering for the original DAG. Its scheduling length is 11. If all the tasks are in the same cluster, then the scheduling length is 12. So figure3(d) is an optimal clustering.

This example not only shows that nonlinearly clustering is better than a linearly clustering for a fine grain DAG, but also shows that the finer the granularity of the DAG, the more edges should be in the same cluster.

VII. Concluding Remarks

In this paper, we firstly analyze the many different definitions on the grain of the tasks and the granularity of the Directed Acyclic Graph (DAG), some of them are very useful in the DAG scheduling theory; Secondly, this paper presents a new granularity definition of the DAG which is very powerful for the large communication delay DAGs. Based on this new definition, we proves that nonlinearly clustering is better for linearly clustering for the fine grain DAG via the fork/join sets; Thirdly, we give the bounds for linearly clustering for the DAGs, which is very useful in the practical use; Finally, we give an example to show the benefits of our new definition.

Next, we will try to prove that for the fine grain DAGs, the nonlinearly clustering is better for the linearly clustering for the general DAGs but not limited to the fork/join sets. And we want to see whether there are any bounds for the nonlinearly clustering under the fork/join granularities.

References

[1] R.L. Graham. "Bounds on Multiprocessing Anomalies and Related Packing Algorithms." *AFIPS Conf. Proc.*,*V*40,AFIPS Press,Montvale,N.J.,1972:205-217.

[2]B.Kruatrachue and T.Lewis. "Grain Size Determination for Parallel Processing." *IEEE Software*,Jan..1988:23-32.

[3]Apostolos Gerasoulis and Tao Yang, "On the Granularity and Clustering of Directed Acyclic Task Graphs.", *IEEE Transactions on Parallel and Distributed Systems*,1993,V4(6):686-701

[4]C. McCreary and H.Gill. "Automatic Determination of Grain Size for Efficient Parallel Processing", *Communication of the ACM*.V32(9),1989:1073-1078

[5] Yiqun Ge and David Y.Y. Yun. "A Method that Determines Optimal Grain Size and Inherent Parallelism Concurrently", *Proceedings of the 1996 International Symposium on Parallel Architectures, Algorithms and Networks*(ISPAN'96):200-206

[6] Apostolos Gerasoulis, Sesh Venugopal and Tao Yang. "Clustering Task Graphs for Message Passing Architectures", *Proceedings 1990 International Conference on Supercomputing*:447-456

[7]Sarkar V. "Partitioning and Scheduling Parallel Programs for Execution on Multiprocessors". PHD thesis, Stanford University,1989

[8]A. Gerasoulis and S. Venugopal. "Linear Clustering of Linear Algebra Task Graphs for Local Memory Systems.", Preliminary Report.

[9]A.A. Khan, C. L. McCreary, M. S. Jones. A Comparison of Multiprocessor Scheduling Heuristics, International Conference on Parallel Processing, 1994

[10]Tao Yang and Aposolos Gerasoulis. "DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors", *IEEE Transactions on Parallel and Distributed Systems*,V5(9): 951-967

[11]Emmanuel Jeannot. "Automatic Code Generation in the Task Graph Model", Research Report, Universite Bordeaux, France,2000

[12]Michael A.Palis,Jing-Chiou Liou and David S.L.Wei: Task clustering and scheduling for distributed memory parallel architectures. *IEEE Transactions on Parallel and Distributed Systems*,7(1):46-55,1996