Mapping single and multiple multilevel structures onto the hypercube

S.G. Ziavras

Indexing terms: Hypercube, Multilevel structures, Parallel computing

Abstract: The paper introduces algorithms that map single and multiple multilevel structures onto the hypercube. For the case of the pyramid, which is a special multilevel structure, it is shown that a new algorithm is a compromise among existing algorithms with regard to cost and performance. Comparative analysis of the algorithms is carried out using analytical techniques and simulation results.

1 Introduction

The hypercube topology has been widely used in parallel computing because of its small diameter, high degree of fault tolerance, and rich interconnection structure [1, 2]. Because of these properties, the hypercube can efficiently emulate a large variety of topologies, such as rectangular meshes, trees, pyramids, etc. [2, 3]. The k-dimensional hypercube H_k contains 2^k nodes. If unique k-bit addresses are assigned to its nodes, an edge connects two nodes if, and only if, their addresses differ by a single bit.

The pyramid topology is composed of successive layers of mesh-connected two-dimensional arrays, where the size of the arrays decreases with the increase of the level number (the base is level 0). Each node, except for nodes at the lowest level, is directly connected to four children at the immediately lower level and the size of each array is 1/4 the size of the array at the immediately lower level [4]. The pyramid P_n with $2^n \times 2^n$ nodes at its base comprises $n + 1$ levels. The pyramid is appropriate for low- and intermediate-level computer vision [4–6]. Multilevel structures are more general than pyramids. The reductions between pairs of neighboring levels are $2^m \times 2^m$, where m is a natural number, and the reductions between different pairs of neighboring levels may differ [6, 7].

This paper proposes algorithms that map single and multiple multilevel structures onto the hypercube. These algorithms yield high performance through high utilisation of PEs (i.e. processing elements composed of a processor and local memory). For the special case of the pyramid, the performance of an algorithm proposed here is compared with the performance of algorithms presented by Stout [5], and Lai and White [8]. Only pyramid mappings have been proposed in the literature.

Paper 9336E (C1), first received 13th April and in revised form 26th October 1992

The author is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, 323 Martin Luther King Boulevard, University Heights, Newark, New Jersey 07102, USA

2 Mapping single pyramids

The cost of graph mappings can be evaluated through the incorporation of three performance measures, namely expansion, dilation, and congestion [8]. Let the function $h: G \rightarrow G'$ represent a one-to-one or many-to-one mapping of the graph G onto the graph G'. The three measures are defined as follows:

Expansion: The expansion of h is the ratio of the size of $V(G')$ to the size of $V(G)$ (i.e. $|V(G')|/|V(G)|$), where $V(G)$ and $V(G')$ are the vertex sets of G and G', respectively. When $|V(G')| \geqslant |V(G)|$, the expansion measures how much of G' is not directly used in the mapping of G.

Dilation: The dilation of an edge in G is the length of the corresponding path in G'. Large dilation results in large communication times owing to longer paths.

Congestion: The congestion is the number of edges in G that share the same edge in G'. Large congestion results in large communication times caused by waiting queues in intermediate nodes.

The remainder of this Section discusses briefly three existing algorithms that map a pyramid onto a hypercube. The first algorithm, which was proposed by Stout, maps the P_n pyramid onto the H_{2n} hypercube [5]. As the pyramid with base $2^n \times 2^n$ contains $\lfloor 2^{2(n+1)}/3 \rfloor$ nodes, the expansion is less than 1. Nodes from the base of the pyramid are mapped onto PEs of the hypercube as follows. The n-bit reflected Gray code is used to encode separately the rows and columns. The binary addresses of hypercube PEs are found by concatenating the bits of the encoded row and column numbers. This yields perfect mapping for the base (i.e. all three measures of cost are optimal). Each node at the next level of the pyramid has four children at the base; therefore one PE from each square of four PEs is chosen to emulate the parent node. Squares marked with 1 in Fig. 1 represent level 1 PEs for the P_3 pyramid and the H_6 hypercube. PEs which have the least-significant bit of their encoded row and column numbers equal to 0 are chosen to represent level 1 nodes of the pyramid. In general, PEs which have the lower k bits of their encoded row and column numbers equal to 0 emulate nodes from level k of the pyramid. Dilations corresponding to lateral connections are equal to 1 for all levels. The maximum dilation is equal to 2 and corresponds to pairs of parents and some children. As a single

This paper is based upon work supported by the National Science Foundation under grant CCR-9109084. The author would like to thank S.C. Patel for his work on the simulation results described in this paper.

	000	001	011	010	110	111	101	100
000	1	2	3	1	1		2	1
001	1′	2′	3′	1′	1′		2′	1′
011	1′			1′	1′			1′
010	1			1	1			1
110	1			1	1			1
111	1′			1′	1′			1′
101	1′	2′		1′	1′		2′	1′
100	1	2		1	1		2	1

Fig. 1 *Mapping P_3 pyramids onto the H_6 hypercube*

PE may emulate a number of pyramid nodes from different levels (for example, the PE with row number 0 and column number 0 emulates nodes from all levels of the pyramid), the hypercube is not capable of emulating multiple levels of the pyramid simultaneously. This mapping is efficient if the pyramid algorithm proceeds level by level. However, algorithms that keep multiple levels of the pyramid active simultaneously are often very important [9].

Two mapping algorithms of high complexity were suggested by Lai and White [8]. They map distinct nodes of the pyramid onto distinct PEs of the hypercube and maintain minimal expansion. They require an H_{2n+1} hypercube for the mapping of the P_n pyramid. The first algorithm has maximum congestion 2 and maximum dilation 3 while the second algorithm has maximum dilation 2 and maximum congestion 3.

In contrast to Stout's algorithm, the latter two algorithms allow all levels of the pyramid to be active simultaneously. Nevertheless, they need double the number of PEs required by Stout's algorithm (i.e. 2^{2n+1} compared with 2^{2n}). In addition, the maximum dilations are 3 and 2 compared to 2 in Stout's algorithm, and the maximum congestions are 2 and 3, respectively, compared to 2 in Stout's algorithm. Even though Stout did not refer to the maximum congestion, it is equal to 2 because some edges are used for communications between siblings and also between parents and children. Owing to increased dilation or congestion, the communication delay in Lai–White's algorithms may be higher when compared to Stout's algorithm.

3 Mapping single/multiple multilevel structures

This Section proposes algorithms that map single and multiple multilevel structures with bases $2^n \times 2^n$ onto the H_{2n} hypercube. Thus, the expansion in all these cases is less than 1. As already discussed in the Introduction, the pyramid is a special case of a multilevel structure [6]. Two algorithms are proposed for the mapping of a single multilevel structure. For reduction $2^m \times 2^m$ between levels 0 and 1, the first algorithm maps level 1 nodes similarly to Stout's algorithm for level m nodes of the pyramid. Generally, for a multilevel structure with reductions $2^{m_i} \times 2^{m_i}$ between levels $i-1$ and i, the mapping of level j nodes is identical to that for level $\sum_{i=1}^{j} m_i$ nodes of the pyramid. The maximum dilation for the pair of levels $i-1$ and i is equal to $2m_i$. Like Stout's algorithm, this mapping algorithm does not support the concurrent emulation of multiple levels.

The second algorithm maps a single multilevel structure with base $2^n \times 2^n$ onto the H_{2n} hypercube so that as many levels as possible are active simultaneously. We first present its application for pyramids. For perfect mapping of the base, the binary reflected Gray code is used as in Stout's algorithm to encode rows and columns. PEs of the hypercube chosen to emulate parents of these nodes correspond to encoded column and row numbers which have their least-significant bits equal to 0 (as in Stout's algorithm). For each set of four PEs that represent siblings at level 1 with common parent, the PE chosen to serve as the parent is neighbour to one of these PEs, and all parent PEs form mirror images in squares outlined by the children. This procedure is repeated until the apex of the pyramid is reached. PEs marked with 1, 2 and 3 in Fig. 1 emulate nodes from levels 1, 2 and 3, respectively, of the P_3 pyramid. As these PEs are distinct, multiple levels of the pyramid, not including the base, can be active at the same time. The maximum dilation of the mapping for pairs of PEs from levels 0 and 1 is 2 and the maximum congestion is 2 (similar to Stout's algorithm). The maximum dilation and maximum congestion for higher levels are 3 and 2, respectively. Thus, this algorithm is a compromise between Stout's algorithm and Lai–White's algorithms with respect to cost and performance for algorithms that activate multiple levels of the pyramid simultaneously. The remainder of the paper analyses only the second of the two proposed algorithms, because it is the one that differs from the other existing algorithms in the case of pyramid mapping.

The extension of the latter algorithm for a multilevel structure with reductions $2^{m_i} \times 2^{m_i}$ between levels $i-1$ and i suggests that for the mapping of level j nodes a subset of PEs that emulate level j nodes of the pyramid be used. These PEs will be as far apart as possible in the $2^n \times 2^n$ array. The maximum dilation for parents and children at levels i and $i-1$, respectively, is equal to $2m_i + 1$ if $i > 1$, and $2m_1$ if $i = 1$. The maximum congestion is still 2.

According to Stout's mapping algorithm, one of the four PEs in each square at the base represents a parent of the immediately higher level. The remaining three PEs in each square could be used to concurrently emulate three more pyramids with the same base and dilation. In contrast, the new algorithm can emulate only two such pyramids simultaneously. PEs marked with prime numbers in Fig. 1 emulate the second pyramid.

4 Simulation results

The following times expressed in machine cycles are assumed. The time it takes to load the values of pixels into the corresponding PEs at the base is equal to 2. Each PE receives one pixel. The addition of two values takes 1 time unit, whereas the multiplication of two values takes 2 time units. Finally, the transmission time for a single value is equal to 2 while the setup time to receive or transmit a single value is equal to 1.

Table 1 shows simulation results for an algorithm that finds the perimeter of an object in an image. It assumes that the boundary pixels are known. A bottom-up process is applied to count the total number of boundary pixels. The results show that Lai–White's algorithms perform the same as the new algorithm for the perimeter counting problem, with respect to the total execution time, if only one level of the pyramid is active at a time. The same table also contains results for pipelining, where there is a need for all levels of the pyramid to be active at the same time to find the perimeter of multiple objects. The new algorithm performs a little poorer than Lai–White's algorithms in the latter case because the base is not allowed to be active by the new algorithm during concurrent multilevel operations. However, the average utilisation of PEs is higher for the new algorithm because Lai–White's algorithms do not use all of the PEs.

Table 1: Perimeter counting

Lai–White's algorithm 1; H_{2n+1}							
One level active			Multiple levels active			n	PEs
Ex. time	Utilisation		1/throughput	Utilisation			
	Avg.	Max.		Avg.	Max.		
40	8.14	15.00	13	25.06	46.15	3	128
53	6.26	11.32	13	25.50	46.15	4	512
66	5.08	9.09	13	25.78	46.15	5	2048
79	4.21	7.59	13	25.80	46.15	6	8192
92	3.62	6.52	13	25.81	46.15	7	32768
Lai–White's algorithm 2; H_{2n+1}							
40	7.40	15.00	13	22.77	46.15	3	128
53	5.64	11.32	13	23.03	46.15	4	512
66	4.54	9.09	13	23.05	46.15	5	2048
79	3.80	7.59	13	23.07	46.15	6	8192
92	3.26	6.52	13	23.07	46.15	7	32768
Stout's algorithm; H_{2n}							
28	14.12	21.43				3	64
37	10.78	16.22				4	256
46	8.69	13.04				5	1024
55	7.27	10.90				6	4096
64	6.25	9.37				7	16384
New algorithm; H_{2n}							
40	9.88	15.00	14	28.23	42.85	3	64
53	7.52	11.32	14	28.48	42.85	4	256
66	6.06	9.09	14	28.55	42.85	5	1024
79	5.06	7.59	14	28.56	42.85	6	4096
92	4.35	6.52	14	28.57	42.85	7	16384

Table 2 shows simulation results for an algorithm that applies two-dimensional convolution. The window sizes for the convolution are 3 × 3, 5 × 5 and 8 × 8, and the numbers of levels in the pyramids are 3, 4 and 4, respectively. The convolution algorithm involves sequences of lateral communications, internal PE operations, bottom-up communications and top-down communications. Several operations can be pipelined. For the convolution problem, the new algorithm performs better than Stout's algorithm when multiple levels of the pyramid are active at the same time. This is because, in contrast to Stout's algorithm, multiple levels may be active simultaneously for the new algorithm. Thus, when the execution times at each level of the pyramid are large enough to take advantage of the parallel execution, the new algorithm will perform better than Stout's algorithm at the same cost. Similarly to the results for the perimeter counting problem, the utilisation of PEs is higher for the new algorithm when compared to the other algorithms.

Table 2: Convolution

Lai–White's algorithm 1; H_{2n+1}						
One level active			Multiple levels active			window
Ex. time	Utilisation		1/throughput	Utilisation		
	Avg.	Max.		Avg.	Max.	
352	8.08	12.10	235	12.78	19.15	3 × 3
976	7.64	12.80	651	11.45	19.20	5 × 5
2497	7.49	12.82	1665	11.23	19.22	8 × 8
Lai–White's algorithm 2; H_{2n+1}						
352	9.07	13.59	235	12.78	19.15	3 × 3
976	7.61	12.80	651	11.42	19.20	5 × 5
2497	7.49	12.82	1665	11.21	19.22	8 × 8
Stout's algorithm; H_{2n}						
244	15.11	18.44				3 × 3
679	14.30	18.55				5 × 5
1729	14.11	18.50				8 × 8
New algorithm; H_{2n}						
244	15.11	18.44	163	22.62	27.60	3 × 3
679	14.30	18.55	451	21.53	27.93	5 × 5
1729	14.11	18.50	1153	21.16	27.74	8 × 8

Table 3 shows simulation results for an image segmentation algorithm that applies a cooperative, iterative approach to segmentation [10]. The algorithm applies sequences of bottom-up and top-down processes for approximately a dozen iterations. Like the other two problems, the new mapping algorithm performs better than Stout's algorithm and reasonably well when compared to Lai–White's algorithms when multiple levels of the pyramid are active at the same time.

Table 3: Segmentation

Lai–White's algorithm 1; H_{2n+1}							
One level active			Multiple levels active			n	PEs
Ex. time	Utilisation		1/throughput	Utilisation			
	Avg.	Max.		Avg.	Max.		
1451	16.29	26.19	700	33.77	54.29	3	128
2181	10.84	17.42	700	33.77	54.29	4	512
2911	8.51	13.05	700	35.39	54.29	5	2048
3641	6.89	10.44	700	35.81	54.29	6	8192
4371	5.75	8.69	700	35.92	54.29	7	32768
Lai–White's algorithm 2; H_{2n+1}							
1291	15.85	29.43	620	33.01	61.29	3	128
1941	10.82	19.58	620	33.86	61.29	4	512
2591	8.35	14.67	620	34.88	61.29	5	2048
3241	6.72	11.72	620	35.13	61.29	6	8192
3891	5.61	9.77	620	35.19	61.29	7	32768
Stout's algorithm; H_{2n}							
1451	24.47	37.22				3	64
2181	15.55	24.76				4	256
2911	11.78	18.55				5	1024
3641	9.45	14.83				6	4096
4371	7.39	12.35				7	16384
New algorithm; H_{2n}							
1451	24.47	37.22	710	50.00	76.06	3	64
2181	15.54	24.76	710	48.45	76.06	4	256
2911	11.78	18.55	710	48.31	76.06	5	1024
3641	9.45	14.83	710	47.74	76.06	6	4096
4371	7.39	12.35	710	45.48	76.06	7	16384

5 Conclusions

Algorithms have been proposed in this paper that map single and multiple multilevel structures onto hypercubes. As the pyramid is a special case of a multilevel structure, these algorithms are also applicable to pyramid mappings. For single pyramid mappings, a new algorithm is a compromise between Lai–White's algorithms and Stout's algorithm with respect to cost and performance. Like Stout's algorithm, the new algorithm requires an H_{2n} hypercube for the mapping of the P_n pyramid, while Lai–White's algorithms require an H_{2n+1} hypercube. Lai–White's algorithms may perform slightly better than the proposed algorithm when all levels of the pyramid are active at a time. However, the new algorithm performs better than Stout's algorithm for concurrent multilevel tasks because it allows for simultaneous execution at multiple levels, while in Stout's algorithm only a single level may be active at a time.

6 References

1 SEITZ, C.L.: 'The cosmic cube', *Commun ACM*, 1985, **28**, (1), pp. 22–33

2 ZIAVRAS, S.G.: 'On the problem of expanding hypercube-based systems', *J. Parallel Distrib. Comput.*, 1992, **16**, pp. 41–53

3 HO, C.-T., and JOHNSSON, S.L.: 'Dilation d embedding of hyper-pyramid into a hypercube'. Proceedings of the Supercomputing '89 Conference, Reno, Nevada, USA, Nov. 13–17, 1989, pp. 294–303

4 CANTONI, V., and LEVIALDI, S.: 'Multiprocessor computing for images', *Proc. IEEE*, 1988, **76**, (8), pp. 959–969

5 STOUT, Q.F.: 'Hypercubes and pyramids', *in* CANTONI, V., and LEVIALDI, S. (Eds.): 'Pyramidal systems for computer vision'. Springer-Verlag, 1986, pp. 74–89

6 ZIAVRAS, S.G.: 'On the mapping problem for multi-level systems'. Proceedings of the Supercomputing '89 Conference, Reno, Nevada, USA, Nov. 13 17, 1989, pp. 399–408

7 ZIAVRAS, S.G.: 'Efficient mapping for a class of hierarchical systems', *IEEE Trans. Parallel Distrib. Syst.*, in press

8 LAI, T.-H., and WHITE, W.: 'Mapping pyramid algorithms into hypercubes', *J. Parallel Distrib. Comput.*, 1990, **9**, pp. 42–54

9 CLERMONT, P., and MERIGOT, A.: 'Real time synchronization in a multi-SIMD massively parallel machine'. Proceedings of Architectures for Pattern Analysis and Machine Intelligence Conference, 1987, pp. 131–136

10 BURT, P.J., HONG, T.-H., and ROSENFELD, A.: 'Segmentation and estimation of image region properties through cooperative hierarchical computation', *IEEE Trans. Syst. Man Cybern.*, 1981, **11**, (12), pp. 802–809

11 CHAN, T.F., and SAAD, Y.: 'Multigrid algorithms on the hypercube multiprocessor', *IEEE Trans. Comput.*, 1988, **C-35**, (11), pp. 969–977