Journal of Interconnection Networks
Vol. 8, No. 3 (2007) 285–304

World Scientific
www.worldscientific.com

AN EMPIRICAL STUDY OF HIERARCHICAL DIVISION FOR MESH-STRUCTURED NETWORKS

DAJIN WANG
Department of Computer Science
Montclair State University
Upper Montclair, NJ 07043, USA
wang@pegasus.montclair.edu

Received 10 February 2006
Accepted 2 October 2007

Abstract

A parallel/distributed system consists of a collection of processes, which are distributed over a network of processors, and work in a cooperative manner to fulfill various tasks. A hierarchical approach is to group and organize the distributed processes into a logical hierarchy of multiple levels to achieve better system performance. It has been proposed as an effective way to solve various problems in distributed computing, such as distributed monitoring, resource scheduling, and network routing. In [21], we studied hierarchical configuration for mesh and hypercube networks to the end of achieving better system performance. In particular, we proposed theoretically optimal hierarchy for mesh and hypercube, so that the total traffic flow over the network is minimized.

In this paper, we present the experimental results to establish the practical relevance of mesh hierarchy proposed in [21]. We simulated situations for multi-level division, real network loading scenarios, random data aggregation rates, and different division sizes other than derived in [21]. The simulation results not only show that the analytically obtained hierarchy works well for many realistic settings, but also offer some useful insights into the proposed hierarchy scheme.

Keywords — Hierarchical architecture, Hierarchy, Interconnection networks, Mesh, Parallel and distributed systems, Simulation.

1 Introduction

The topologies of many distributed systems are more or less hierarchical. If distributed functions are performed in such a way as to reflect the underlying hierarchical topology, the algorithm design can be simplified. A hierarchical architecture can also improve scalability of the distributed functions and optimize their performance by increasing parallelism and reducing information flow. As a matter of fact, the hierarchical approach that allows distributed processes to operate on a logical structure is an established design methodology, and has been used in a variety of forms for solving different distributed control problems, such as distributed monitoring, resource scheduling, and network routing, either to effectively coordinate the local control activities or to enhance the overall performance [1, 2, 5, 6, 8, 20].

In [21], we proposed hierarchical schemes for two regular networks: mesh and hypercube. Both are very popular networks, have been extensively studied, and commercial parallel computers using them have been available for a long time. We proposed hierarchy schemes on mesh and hypercube that are shown to greatly reduce the traffic flow. As in most cases of optimization, a hierarchical configuration optimizing in all aspects is impossible to achieve. Therefore it is important to characterize the applicability of the proposed scheme. Generalizing the hierarchical monitoring system in previous works [4, 18], the hierarchical configuration we presented in [21] was best applicable to those tasks that need to process data collected from all processors of the network, and the nature of the task allows "partial preprocessing" of data before they reach their final destination. There are many such data in both computational and managerial tasks. A simple example is to get the sum of certain value from all processors: it is not necessary for the master adder of the network to collect all addends before it performs the addition — partial sums can be obtained by some "submasters," and sent to the master adder. That will prevent many pieces of data from traveling all the way to the master, reducing the traffic flow in the network. Under this context, by first limiting the levels of the hierarchy to two, we studied optimal hierarchical configurations for mesh and hypercube. Based on analytical results, partitioning algorithms was presented which were optimal in terms of total communication cost.

In this paper, we present a set of experimental results conducted for mesh network. Experimental simulation is an effective means to evaluate the competence of hierarchical schemes. In cases of irregularly connected networks, or irregularly distributed traffic loads, it is the only means to definitely quantify the improvement brought by a specific hierarchy method. In [21], we tackled the problem of hierarchical configuration for mesh and hypercube in a theoretical framework. For analytical tractability, the model we used to obtain the scheme is a simplification of realistic situations. It is then crucial to understand the practical relevance of the derived hierarchical methods. The work of this paper is to extend the analytical results by performing experimental measurements, using the theoretical result as a guide to simulate hierarchical meshes for a variety

of realistic settings.

The rest of this paper is organized as follows. In Section 2, we describe the hierarchical architecture systems, and formulate the optimal configuration problem under consideration. We then introduce the analytical results of [21], based on which the experiments are carried out. In Section 3, we present a set of simulation results on mesh network for a variety of realistic scenarios, which include multi-level division, non-unified traffic loads, hierarchy performance under various aggregation rates, finding "division-worthy threshold" for different submesh sizes, etc. The implication of the experimental results will be explained and discussed. In Section 4 we summarize the work of this paper and outline possible directions for future work.

2 Analytical model for hierarchical division of mesh-structured networks

The problem of optimal hierarchical configuration of a distributed system is concerned with finding an optimal partition of the processes/processors in a given network environment. A configuration of a system has three components: (1) a hierarchical partition of the nodes (processes/processors), defined by the levels of the hierarchy, (2) the grouping of nodes at each level, and (3) the location of the leader at each group. Optimization means to minimize the total processing cost. The three costs that are of primary consideration are the amount of memory required, the amount of communication between units, and the time required for processing. The work of [21] mainly focused on reducing the amount of communication over the network.

In order to formulate such a problem quantitatively for analysis, in [21] we assumed a simplified model for the process of collecting information and perform some processing (e.g. combining information, making decisions) based on it. In that context, a hierarchical organization consists of local processes which aggregate information from other processes in the hierarchy, passing that information through the hierarchy. During this process, information can be condensed as it passes through the hierarchy. The target was to minimize the total communication cost, i.e., the amount of data flowing in the network. In mesh-connected computers, a node sending data to its group leader has to pass all intermediate nodes. Messages sent between processors far apart have to travel long way, incurring great amount of data flow in the network. We normalized a node's cost for traveling to its immediate neighbor to "one step." The number of steps that a node travels to its leader is said to be its *communication cost*. For example, if there are 4 intermediate nodes between a node and its data collector, the communication cost for collecting this node's data is 5. The total communication cost is the total number of steps that all nodes have to travel in the processing. The work of [21] was to propose hierarchical division schemes for mesh (and hypercube) so that the total communication cost was minimized. It is worth pointing out that in

the treatment of [21], we assumed a normalized "1" cost for each hop of communication for the purpose of mathematical tractability. In this paper, we will extend the analytical result by simulating a variety of realistic scenarios. In particular, we will examine the cases where we dispose of the assumption of unit hop cost. The simulation results therefore will serve the two-fold purpose of checking out the proposed scheme's practical relevance, and giving us the true sense of how much improvement it brings about in reality. In the rest of this section we present the analytical results of [21], based on which the experiments of this paper are conducted.

Let the squared mesh contain N^2 processors, with dimensions $N \times N$. If the whole mesh is viewed as one hierarchy (one-level), then choosing the center node as the master node (i.e., the group leader with the entire mesh as the group) would obviously minimize the total communication cost. Depending on whether N is an odd or even number, the cost can be calculated as follows. See Figure 1. As has been pointed out, a simplified model is used in calculating the system's communication cost: here we assume a unit of packet from each node going to the master node, and for each intermediate node it passes, we count one unit of communication cost. The farther the node is from master, the more cost it incurs to travel to master.

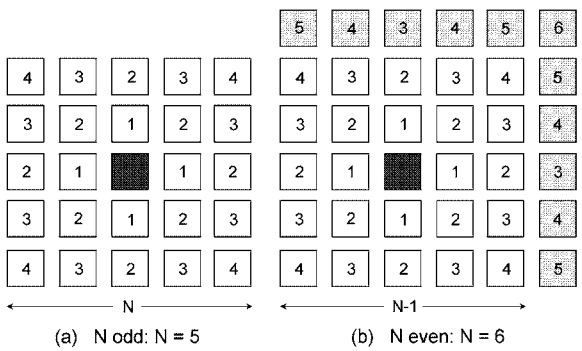

Figure 1: (a) A 5×5 mesh. At center is the master node (dark). Numbers in all other nodes represent their communication costs. (b) A 6×6 mesh. The dark node at the "pseudo center" is the master.

For the scenario that all nodes send a packet to the master, the total communication cost, denoted as $C(N)$, is calculated as

$$C(N) = \begin{cases} \frac{N^3-N}{2}, & N \text{ odd} \\ \\ \frac{N^3}{2}, & N \text{ even} \end{cases} \tag{1}$$

It can be shown that using any non-central monitor would cost more, with the monitor at corner costing most.

In a two-level partitioning, the whole mesh is divided into several submeshes. Each submesh has a group leader. A group leader will collect data in its submesh, and then turn the data in to a leader at the higher level. The purpose of hierarchical division is to reduce the cost incurred by communication. Assuming two-level hierarchy, we need to find out the best way to divide the mesh so that the total communication cost is minimum. Figure 2 illustrates the structure of two-level hierarchical mesh.

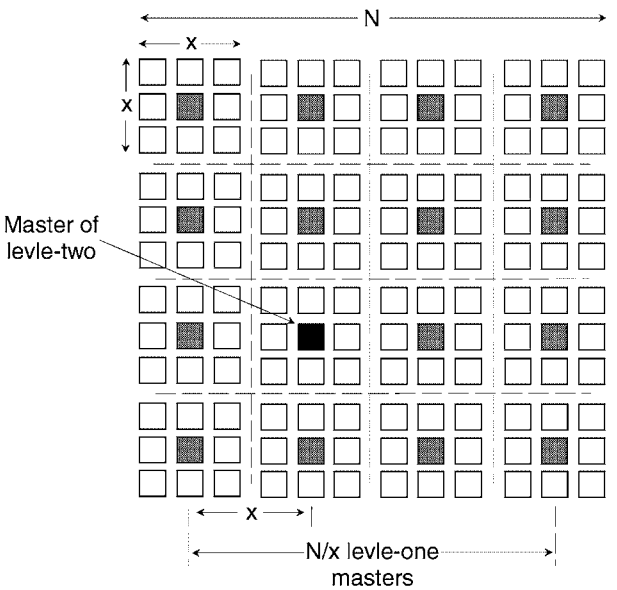

Figure 2: Two-level partitioning. The $N \times N$ nodes are divided into $(\frac{N}{x})^2$ $x \times x$ submeshes. The grey nodes are level-one leaders, the darkest node is level-two leader.

In the two-level hierarchical mesh, data transfer to the master node is performed in two phases. In the first phase, all level-one masters receive data from nodes in their corresponding submeshes. All level-one masters then aggregate and/or preliminarily process the collected data. In the second phase, the level-two master (the darkest node in Figure 2) collects data (that have been processed/aggregated) from all level-one masters. In [21], an optimal submesh size x is calculated, so that the total communication is minimized: for the two-level cost $C_{2L}(N,x)$, taking the derivative $C_{2L}(N,x)'_x$ with respect to submesh size x, and solving $C_{2L}(N,x)'_x = 0$ for x, we obtained an optimal submesh size x:

$$x = \begin{cases} \frac{\sqrt[3]{27N+3\sqrt[2]{3+81N^2}}}{3} - \frac{1}{\sqrt[3]{27N+3\sqrt[2]{3+81N^2}}}, & x \text{ odd} \\ \sqrt[3]{2N}, & x \text{ even} \end{cases}$$

The difference between $\sqrt[3]{2N}$ and $\left(\frac{\sqrt[3]{27N+3\sqrt[2]{3+81N^2}}}{3} - \frac{1}{\sqrt[3]{27N+3\sqrt[2]{3+81N^2}}}\right)$ is vanishingly small. So for all practical purposes, we can just use an integer close to $\sqrt[3]{2N}$ for the size of submesh to achieve the minimum total cost.

Theorem 1 [21] *In a two-level hierarchical mesh, if the level-1 submesh is of dimensions $x \times x$, so that*

1. *x is as close to $\sqrt[3]{2N}$ as possible*
2. *x divides N*

then the system's total communication cost is minimum.

Figure 3 illustrates an example of level-1, level-2, and total costs as function of submesh size x, for a two-level hierarchical mesh, where the original mesh size is $N = 72$. $\sqrt[3]{2N} = \sqrt[3]{144} \approx 5.24$. According to Theorem 1, $x = 6$ will be chosen as the optimal submesh size. The total cost is 20736, which is minimum.

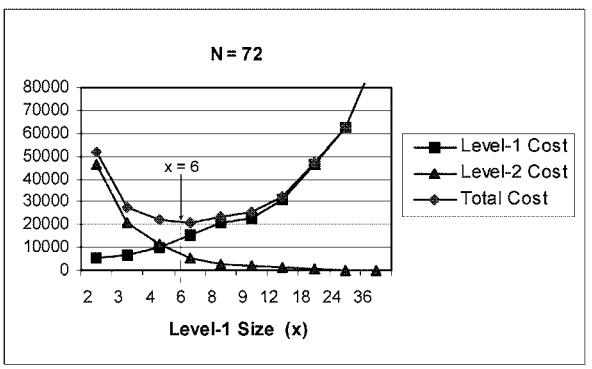

Figure 3: Level-1, level-2, and total costs as function of submesh size x. The original mesh size is $N = 72$. It can be seen that there is a minimum total (two-level) cost.

The saving of communication cost gained by this two-level hierarchical scheme is quite substantial. The ratio of min.-two-level-cost/one-level-cost is (assuming even N, even x)

$$\frac{\left.\frac{N^2x^3+N^3}{2x^2}\right|_{x=\sqrt[3]{2N}}}{\left(\frac{N^3}{2}\right)} = \frac{3}{2}\sqrt[3]{\frac{2}{N^2}}$$

which is ever decreasing as N grows. Figure 4 shows a comparison between minimum two-level cost and one-level cost: when $N = 10$, the min.-two-level-cost/one-level-cost ratio is about 40%; when $N = 100$, less than 9%; when $N = 200$, less than 6%.

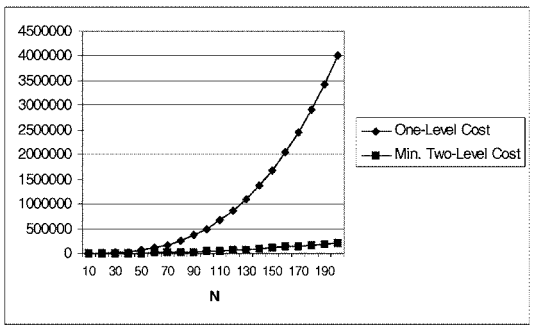

Figure 4: Comparison of minimum two-level cost and one-level cost.

If the levels of partition are more than 2, cost can be further reduced. Extending the approach for two-level partition, for a partition of m levels, let x_1 be the submesh size at level-1 (the bottom level), x_2 the submesh size at level-2, ..., so that $N = x_1 \times x_2 \times \cdots \times x_m$. Ideally, an optimal total cost can be obtained by finding the minimum value of a multi-variable cost function $C(x_1, ..., x_m)$. However, even for a modest m, that is a mathematically burdensome task. So instead of computing the absolutely optimal partition for a larger m, we resort to repeatedly applying the result for two-level partition, until the desired level number is reached or desired total cost obtained.

It is observed that in optimal two-level partition, the cost of level-one (bottom level) is twice as large as that of level-two. Assuming even N, even x:

$$C_I(N, x = \sqrt[3]{2N}) = \frac{1}{2}(x^3) \cdot (\frac{N}{x})^2 \Big|_{x=\sqrt[3]{2N}} = \frac{1}{2}\sqrt[3]{2N^7}$$

and

$$C_{II}(N, x = \sqrt[3]{2N}) = \frac{1}{2}(\frac{N}{x})^3 \cdot x \Big|_{x=\sqrt[3]{2N}} = \frac{1}{4}\sqrt[3]{2N^7} = \frac{C_I}{2}$$

So, for three-level partition, we can do another two-level partition for all submeshes at the bottom level, further reducing the total cost. As the number of levels increases, it is not always the bottom level that has the largest cost. However, keeping track of the level of largest cost is not a difficult job. All we need to do is to compare the current largest cost with the cost of newly obtained level. The heuristic algorithm for multiple-level partition is described in Figure 5. The experimental results regarding multiple-level partition will be presented in Section 3.

An important observation is that the magnitude of cost reduction, in terms of percentage, drops rather quickly as each new level is added, i.e., the most substantial saving occurs when the system goes from one-level to two-level, but much less substantial going from two-level to

```
Do an optimal two-level partition                          /* initial step */
levelCount ← 2
levelOfLargestCost ← 2
repeat
{
    At levelOfLargestCost, do an optimal two-level partition
    Update levelOfLargestCost
    levelCount++
}
until ( levelCount = targetLevelNumber || totalCost ≤ targetCost )
```

Figure 5: Heuristic algorithm for multiple-level partition.

three-level, and so on. Take the example of $N = 72$ again. The ratio of min.-two-level-cost/one-level-cost $\frac{3}{2}\sqrt[3]{\frac{2}{N^2}}\Big|_{N=72} \approx 11\%$, with the optimal level-one submesh size 6, and total cost 20736. The costs at level-1 and level-2 are 15552 and 5184, respectively. Another two-level partition at bottom-level will reduce the cost from 15552 to 8640. So the reduction rate from two-level to three-level is $(8640 + 5184)/20736 \approx 67\%$.

What this observation suggests is that after a few partitions, there will be no much significant gain in cost reduction. Therefore, it is a good idea to set the number of levels to a modest value, such as 3, 4, or 5, for most meshes, directly reducing the time complexity of the algorithm.

3 The experimental study

Appropriately designed experimental simulations are useful tools in evaluating various aspects of a proposed scheme. Especially in cases where an analytical model is elusive, the competence and practical relevance of the proposed scheme can only be assessed by experimental means. Typical examples of such cases include irregularly connected networks, irregularly distributed traffic loads, etc. In this section, we will present and discuss the implication of a series of simulation experiments.

The simulations presented in this section are designed based on the theoretically obtained submesh size, i.e., a factor in the closest neighborhood of $\sqrt[3]{2N}$, which assumes unit transfer of

data in all submeshes and levels. While there are applications fitting in with this assumption, to many applications this is an oversimplification of the real situation. In the experiments, we simulated a variety of more realistic data transfer patterns using the N's close-to-$\sqrt[3]{2N}$ factor as submesh size.

3.1 Multi-level division

According to the analysis at the end of Section 2, the traffic reduction gain dramatically drops as division levels increase. Therefore in the experiments we are mostly interested in investigating network traffic reduction for two-level division. However, we still conducted simulation up to 6-level division to reinforce the analytical conclusion for multi-level division. The experiment results presented in this subsection for multi-level division still assume unit data transfer. Nevertheless, these results suffice to give us the sense that beyond 2-level division, more gains may not worth the overhead incurred by adding more levels.

	Total Costs						**Ratios**				
N	1 Level	2 Levels	3 Levels	4 Levels	5 Levels	6 Levels	2 L / 1 L	3 L / 2 L	4 L / 3 L	5 L / 4 L	6 L / 5 L
4	32	24					75.00%				
6	108	60					55.56%				
8	256	128					50.00%				
9	360	144					40.00%				
10	500	220					44.00%				
16	2048	640	512	480			31.25%	80.00%	93.75%		
32	16384	3072	2560	1920			18.75%	83.33%	75.00%		
64	131072	16384	14336	7680			12.50%	87.50%	53.57%		
72	186624	20736	13824	9600			11.11%	66.67%	69.44%		
128	1048576	81920	49152	40960			7.81%	60.00%	83.33%		
150	1687500	114300	84300	41640			6.77%	73.75%	49.40%		
192	3538944	202752	129024	92160			5.73%	63.64%	71.43%		
200	4000000	222400	142400	99968			5.56%	64.03%	70.20%		
250	7812500	390500	215500	171820			5.00%	55.19%	79.73%		
256	8388608	393216	262144	163840			4.69%	66.67%	62.50%		
320	16384000	665600	460800	256000			4.06%	69.23%	55.56%		
512	67108864	2097152	1572864	655360			3.13%	75.00%	41.67%		
640	131072000	3358720	2211840	1126400			2.56%	65.85%	50.93%		
750	210937500	4921500	3346500	1546820			2.33%	68.00%	46.22%		
1024	536870912	10485760	4718592	3014656	1081344	950272	1.95%	45.00%	63.89%	35.87%	87.88%

Figure 6: Traffic costs and cost ratios with two-, three-, four-, five-, and six-level divisions.

The experiment results are shown in Figure 6. The experiment implements the heuristic multi-level division algorithm outlined in Figure 5. The left half of the table are the total traffic costs when the whole mesh undergoes 2-, 3-, ..., up to 6-level divisions. The numbers of levels the whole mesh can be divided into increases rather slowly as mesh size increases. For example, when N is under 10, the mesh can be divided into at most 2 levels. When N is between 16 and 750, a 4-level division can be worked out. It is not until $N = 1024$ can we obtain a 5- and 6-level division.

The right half of the table shows the ratio of total costs as hierarchy levels grow. For example,

column "3L/2L" is the ratio of the 3-level total cost versus 2-level total cost. The smaller the ratio, the more benefit we gain (in terms of less traffic load) by adding one more level of hierarchy. We can observe that ratios in the "2L/1L" column are consistently much smaller than in all subsequent columns. The data display the pattern that is in accord with the earlier analytical conclusion: the most remarkable, biggest drop in traffic load occurs when the mesh undergoes its first hierarchical division, from no hierarchy to a 2-level structure. The improvement can always be observed as further division takes place. However, the benefits it gains by adding more levels is no comparison to that of 2L/1L.

3.2 Simulation with non-unified traffic loads

We evaluated the proposed scheme's performance by simulating realistic traffic loads. That is, we abandon the assumption of unit data transfer among nodes. In the simulation, we assign variable units of data to be transferred to the central node. See an example of 4×4 mesh in Figure 7.

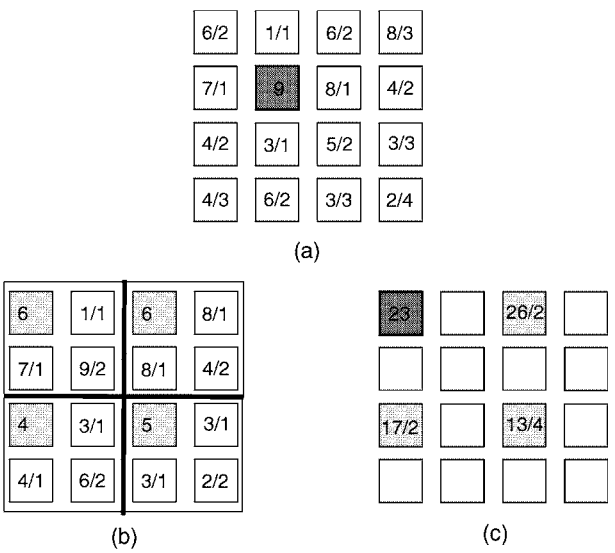

Figure 7: (a) One-level traffic flow. "6/2" means 6 units of data need to travel distance 2; (b) Level-one submeshes, and the corresponding data amount/distance; (c) The level-two mesh and the corresponding data amount/distance.

Figure 7 (a) shows the non-leveled structure, with the darker node being the center. For the numbers "p/q" in each node, p represents the amount of data to be transferred (normalized in a scale of 1 to 10), and q is the distance to the center. Then for this particular example, the total

amount of data that have flown over the transfer channels would be:

$$\sum_{all\ p,q} p \times q = (6 \times 2) + (1 \times 1) + (6 \times 2) + (8 \times 3) + \cdots + (2 \times 4) = 143$$

Figure 7 (b) shows a 2-level hierarchy of the network. The four darker nodes are level-1 masters. With this division, the "q" number now is the distance to a node's level-1 master. Figure 7 (c) shows the level-2 mesh that consists of the 4 level-1 master nodes with the upper-left being the master (and therefore the final "central" node of the network). Note that for non-unified data transfer among nodes, a two-level hierarchy cannot guarantee traffic reduction for all instances — for the given example, the center node has a relatively large amount of data (9), and needs not to be moved in non-hierarchical structure, but needs to be moved around in a two-level hierarchy, incurring more communication costs. Indeed, in the given example, the total communication cost for 2-level structure would be 217 (79 at level-one, 138 at level-two), bigger than the non-hierarchical approach.

However, a crucial, whereas very practical premise for the hierarchical approach is that in many applications, *not all* of the original data need to be transferred to the central node. They could be partially or even entirely processed by their local master nodes. Under this premise, we assign a *data-reduction rate* for all local master nodes. This reduction rate, denote ρ, represents an average percentage of data amount collected by a local master, that is sent to the central node. For the example in Figure 7 again, the amount of data collected at the four local masters are 23, 26, 17, and 13, respectively. If $\rho = 90\%$, then the amount of data transferred to central node (assuming the upper-left node to be the central node) are $26 \times \rho = 23.4$, $17 \times \rho = 15.3$, and $13 \times \rho = 11.7$, respectively. In the example of Figure 7, when $\rho = 90\%$, the 2-level scheme still incurs more communication cost than the non-hierarchical approach. For the hierarchical structure to outperform its non-hierarchical counterpart, the value of ρ must not be higher than 46%. In other words, for this particular example, the hierarchical approach will benefit those applications of which at least 54% of the data (on average) can be processed by local master nodes.

We introduce the notion of *threshold* for quantifying the performance of a hierarchical network: a threshold of a hierarchical network, denoted μ, is a data-reduction rate at which the hierarchical structure starts to outperform its non-hierarchical counterpart. When $\rho = \mu$, the ratio of 2-level/1-level total costs (the 2L/1L value) is 1. Obviously, the higher the μ, the better the hierarchy's performance. We have done simulations to see the relationship between mesh size and μ. The simulation results are shown in Figure 8 and Figure 9. Figure 8 depicts the 2L/1L ratios when adopting various ρ's. The submesh size is an integer that is close to $\sqrt[3]{2N}$ and can divide N. For comparison, the 2L/1L ratio using unit transfer is also shown (the bottom "analytical" curve). (The unit transfer assumption was used when obtaining the $\sqrt[3]{2N}$ submesh size.) From Figure 8,

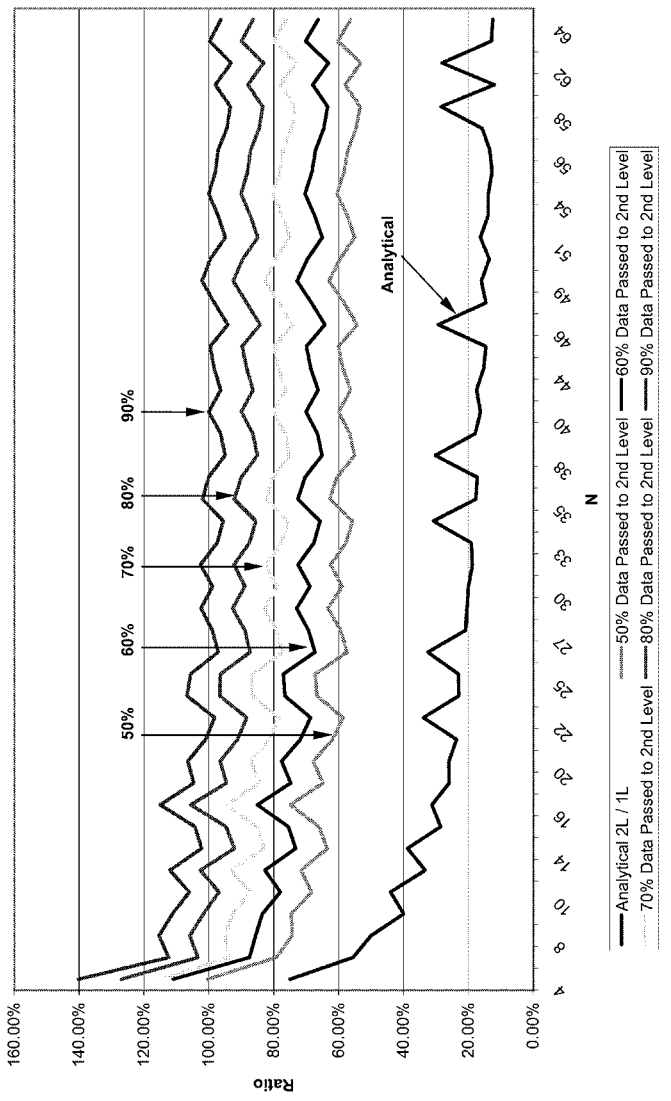

Figure 8: Two-level-cost/one-level-cost ratios with different data-reduction rates.

we can see that when $\rho = 90\%$ and mesh size N is not sufficiently large (< 30), the 2L/1L ratio is above 100%. That means the hierarchical scheme does not pay off when the average transfer rate is too high ($> 90\%$) and N too small. With $\rho = 90\%$ and a larger N, the 2L/1L ratio stays around 100%, but shows a tendency of going down as N gets larger. With a ρ not as high as 90% (say 85%, 80%, 70%, ...), the 2L/1L ratios are all below 100%, meaning the system will benefit from the hierarchical scheme. Also, all curves show the tendency of going down as N increases. That means the larger the mesh size, the more the system benefits from hierarchy, and the incurring overhead pays off better.

Figure 9 shows a direct relationship between the mesh size N and the 2-level hierarchy threshold μ. It can be observed that when N is relatively small, μ is relatively low, which means that hierarchy will benefit only if a relatively large amount of data (between 20% to 50%) can be "absorbed" by the submesh masters. However, when N gets larger (say > 70), the threshold quickly approaches, and stabilizes in between 95% and 100%. That means for large meshes, to benefit from hierarchy, submesh masters only have to absorb less than 5% of data collected in their corresponding submeshes. In other words, even if 95% or more submesh data is transferred to the central node, 2-level approach still can outperform its unleveled counterpart.

3.3 Simulation with random data aggregation

We also simulated the scenario of variable, random reduction rates across all submeshes. Instead of assigning same reduction rate to all submeshes, we randomly assigned 50% to 90% reduction rates to submeshes and observed the resultant average 2L/1L ratios. The findings of the simulation are shown in the chart of Figure 10.

Drawn in Figure 10 is the average result of 100 assignments of random ρ's (between 50% and 90%) among all submeshes. One can see that for relatively small N (say < 30), the 2L/1L ratios are higher than 80%. As N grows, however, the 2L/1L ratios stays in the neighborhood of 80%, and shows a tendency of slowly going down as N becomes very large. Interpreting the simulation result: we can expect about 80% 2L/1L ratio if we adopt the $\sqrt[3]{2N}$ submesh size, and the data-reduction rate among all submeshes are between 50% and 90%. We can expect slow lowering of ratio as N increases.

3.4 Simulation with submesh size smaller than $\sqrt[3]{2N}$

In all simulations reported in preceding subsections, we used the submesh size we obtained analytically: a factor of N in the closest neighborhood of $\sqrt[3]{2N}$. With unified transfer load at all levels, this submesh size would effect the theoretical minimal overall traffic load in the 2-level hierarchical mesh. The preceding subsections presented simulation results of using this submesh

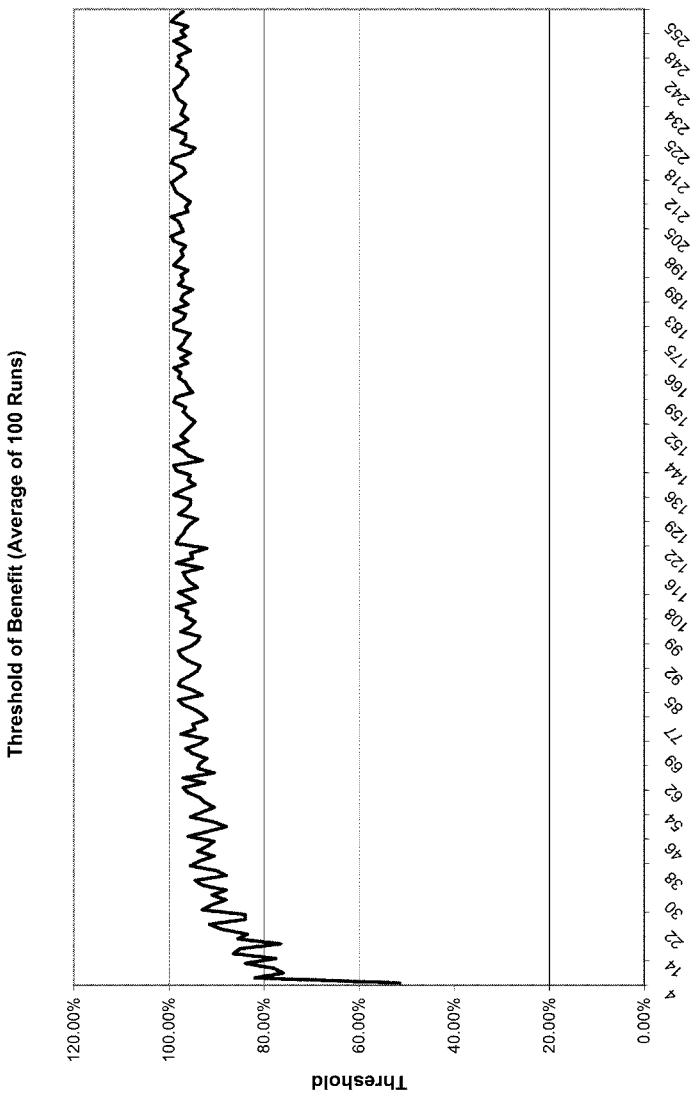

Figure 9: Threshold of benefit as function of N.

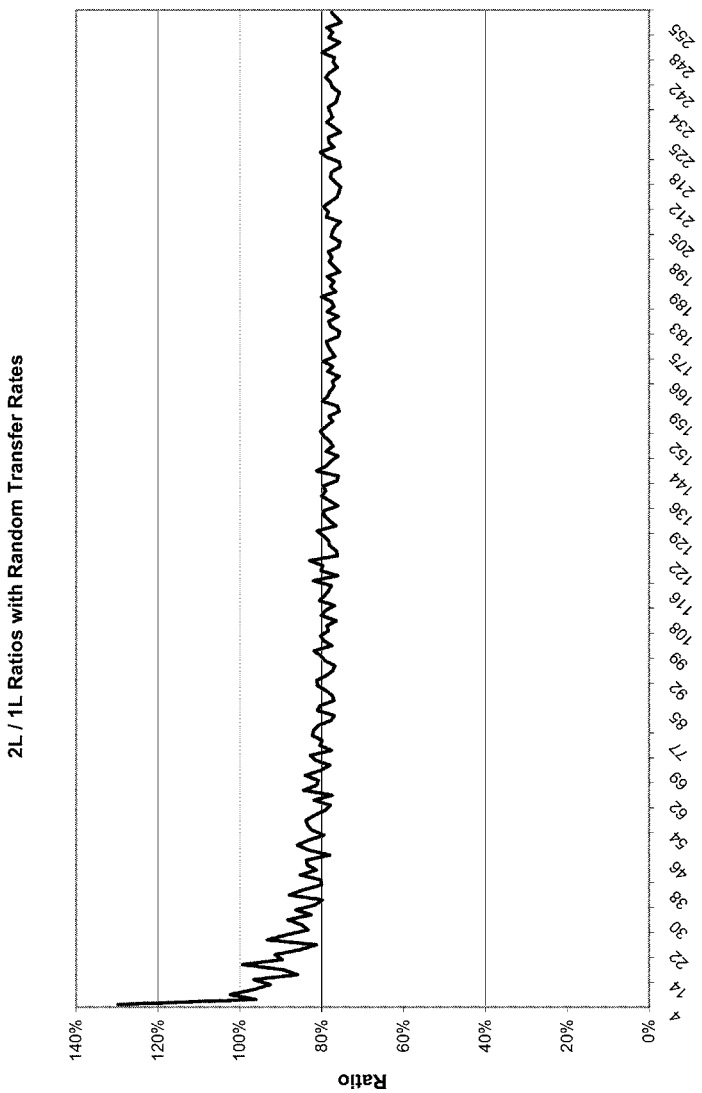

Figure 10: 2L/1L ratios with variable data-reduction rates.

size in a variety of more realistic scenarios. The simulation results show that using this submesh size will bring about noticeable reduction in network traffic load, and the hierarchical scheme is a worthwhile undertaking, especially when the original mesh is large.

It was discovered during the simulation, however, that the traffic load reduction brought about by the hierarchy using the analytically optimal submesh size does not necessarily translate into an absolutely minimal traffic load. The simulation results adopting submesh size smaller than $\sqrt[3]{2N}$ is illustrated in Figure 11.

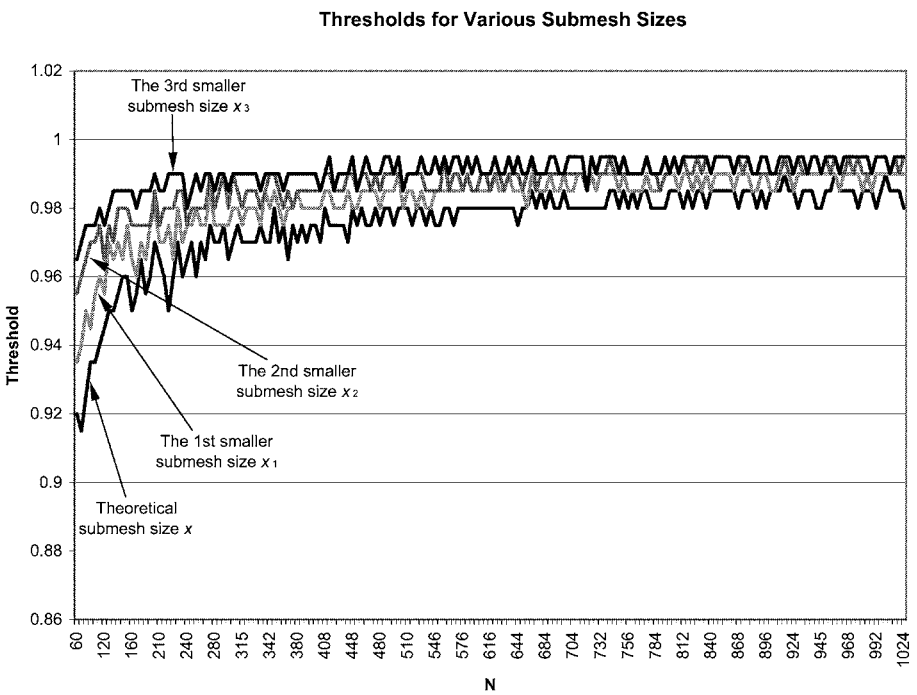

Figure 11: Thresholds for submesh sizes smaller than $\sqrt[3]{2N}$.

In Figure 11, the thresholds for four different submesh sizes are shown. Recall that a threshold is such a pivotal data-reduction rate ρ at the level-2 mesh, that with this ρ the 2L/1L ratio starts to get lower than 100%. We have used μ to denote this particular ρ. A high μ means that the 2-level scheme can outperform its non-leveled counterpart with a high ρ (i.e., for the 2-level scheme to be beneficial, the level-2 mesh only need to absorb a very small portion of the data from submeshes). Hence the higher the μ, the better the hierarchy's performance. In Figure 11, the bottom curve of the four is the threshold as the function of original size using the theoretically derived submesh size x, which is a factor of N in the closest neighborhood of $\sqrt[3]{2N}$.

Going up, the first curve above the bottom one is the threshold function using submesh size x_1, which is the next factor of N smaller than x. The second curve above the bottom one is the threshold function using submesh size x_2, which is the next factor of N smaller than x_1. And so on. We can observe that the threshold curve gets raised as submesh size gets smaller, although the gain is quite small. This means that with non-uniform traffic loads, the theoretically ideal submesh does not necessarily effect the absolutely minimum traffic. Submeshes smaller than $x \times x$ could do better. An intuitive explanation for this observation can be given by referring to Figure 12.

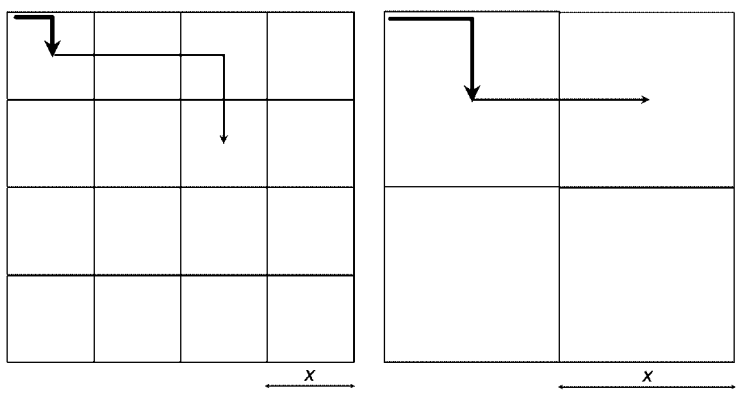

Figure 12: With non-uniform traffic loads, submeshes smaller than $x \times x$ could yield slightly higher thresholds.

The cases of small and large submeshes are shown in Figure 12. The thick arrows represent the internal-submesh, full-scale communication to the submesh center node. The thin arrows represent communication at the higher level, which is of a reduced load. In the case of small submesh, full-scale loads travel *shorter* distance to its center than in large submesh. That is, in a load's journey to the final center, a smaller portion of it is with the full load, and a bigger portion is with reduced load. In the case of large submesh, it's the other way around. This kind of distribution of loads eventually yields a slight gain in threshold.

That is not to say, however, that the smaller the submesh size, the better. This is because in a larger submesh, the local center node controls more nodes than in a smaller submesh. This will enable the local center to better aggregate local data, giving a better chance to lower the data-reduction rate, and eventually lower the overall traffic in the hierarchical network. (For insight, assume hypothetically that the submesh is reduced to contain just one node. Then obviously there is no possibility to do any data-reduction — all data have be transferred at the higher level.) So a balance has to be struck between the submesh size and the threshold it could bring about. Considering that the original x (the closest factor of N in neighborhood of $\sqrt[3]{2N}$) produces

noticeable reduction in traffic in all simulated scenarios, and the threshold gains by N's smaller factors are quite small, we can assert that x can be a leading candidate for submesh size when constructing a 2-level hierarchical mesh.

4 Conclusion

We presented a host of experimental results in an effort to establish the practical relevance of the mesh hierarchy scheme proposed in [21]. The scenarios simulated are multi-level hierarchy, realistic, non-unified network loading, random data aggregation rates, and submesh sizes other than in the closest neighborhood of $\sqrt[3]{2N}$. Through these simulations we hope to get better understanding of the nature of hierarchical meshes, and hopefully help shed more light on the understanding of hierarchical networks in general. The simulation results have shown that the analytically obtained hierarchy scheme works well for many realistic settings. They also offer some useful insights, which are obscured in the process of theoretical calculation, into the proposed hierarchy scheme. From simulation outcome, we can conclude that for a hierarchical mesh-structured network, a good candidate for submesh size is the factor of N in the closest neighborhood of $\sqrt[3]{2N}$, where $N \times N$ is the original mesh dimensions. Even though this particular submesh size is eventually not adopted, it could serve as an appropriate starting point.

Although the hierarchical scheme in this work was proposed with specific context (e.g., the master node of the network performing a task involving data from the all nodes), the approaches developed may be applied to a broader range of hierarchical control/computational problems in distributed processing. Since optimizing network performance using hierarchical levels is a commonly adopted approach in practice, experimental measurements of improvement on real machines/networks would be useful to show the usefulness of this strategy.

Acknowledgment

The author thanks Jason W. Zurawski for implementing the experiments and providing simulation data.

References

[1] D. Agrawal and A. El Abbadi, "An Efficient and Fault-Tolerant Solution to Distributed Mutual Exclusion," *ACM Transactions on Computer Systems*, Vol. 9, No. 1, Feb. 1991, pp.1-20.

[2] I. Ahmad, A. Ghafoor, and G. C. Fox, "Hierarchical Scheduling of Dynamic Parallel Computations on Hypercube Multicomputers," *Journal of Parallel and Distributed Computing*, Vol. 20 (1994), pp.317-329.

[3] J. Cao, O. de Vel, and L. Shi, "Architecture Design of Distributed Performance Monitoring Systems: A Hierarchical Approach," *Proc. 7th International Conference on Parallel and Distributed Computing Systems*, Las Vegas, USA, October 1994, pp. 658-663.

[4] J. Cao, K. Zhang, and O. de Vel, "On Heuristics for Optimal Configuration of Hierarchical Distributed Monitoring Systems," *Journal of Systems and Software*, Elsevier Science Inc., New York. Vol. 43, No. 5, 1998, pp. 197-206.

[5] J. Cao and F. Zhang, "Optimal Configuration in Hierarchical Network Routing," *Proc. 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, Edmonton, Alberta, Canada, May, 1999. pp. 249-254.

[6] C.-H. Edward Chow, "Resource Allocation for Multiparty Connections," *J. System Software*, 1995, Vol. 28, pp. 253-266.

[7] W.J. Dally "Performance Analysis of k-ary n-cube Interconnection Networks," *IEEE Transactions on Computers*, Vol. 39, No. 6, pp. 775-785, June 1990.

[8] G. Feitelson and L. Rudolph, "Distributed Hierarchical Control for Parallel Processing," *Computer*, pp. 65-77, May 1990.

[9] 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.

[10] D. Haban and D. Wybranietz, "A Hybrid Monitor for Behavior and Performance Analysis of Distributed Systems," *IEEE Transactions on Software Engineering*, Vol. 16, No. 2, pp. 197-211, February 1990.

[11] J.K. Hollingsworth and B.P. Miller, "Dynamic Control of Performance Monitoring on Large Scale Parallel Systems," *Proc. International Conference on Supercomputing*, Tokyo, July 1993, pp. 235-245.

[12] J. Joyce, G. Lomow, K. Slind, and B. Unger, "Monitoring Distributed Systems," *ACM Transactions on Computer Systems*, Vol. 5, No. 2, pp. 121-150, May 1987.

[13] M. J. Mataric, "Using Communication to Reduce Locality in Distributed Multi-Agent Learning," *J. Experimental and Theoretical Artificial Intelligence*, Vol. 10, No. 3, 1998. pp.357-369.

[14] B.P. Miller, C. Macrander, and S. Sechrest, "A Distributed Programs Monitor for Berkeley UNIX," *Software – Practice and Experience*, Vol. 16(2), pp. 183-200, February 1986.

[15] O. Ogle, K. Schwan, and R. Snodgrass, "The Real-Time Collection and Analysis of Dynamic Information in Distributed and Parallel Systems," Technical Report, Computer and Information Science Research Center, The Ohio State University, August 1987.

[16] C.-C. Shen and W.-H. Tsai, "A Graph Matching Approach to Optimal Task Assignment in Distributed Computing Systems Using a Minimax Criterion," *IEEE Transactions on Computers*, Vol. C-34, No. 3, pp. 197-203, March 1985.

[17] L. Shi, O. De Vel, J. Cao, and M. Cosnard, "Optimization in a Hierarchical Distributed Performance Monitoring System," *Proc. First IEEE International Conference on Algorithms and Architectures for Parallel Processing*, Brisbane, Australia, April 1995, pp. 537-543.

[18] L. Shi, J. Cao, and O. de Vel, "A Hierarchical, Distributed Monitoring System For Interprocess Communications," *International Journal of Computer Systems: Science and Engineering*, CRL Publishing Ltd. (14), Sep. 1999. pp. 317-325.

[19] M. Spezialetti and J.P. Kearns, "A General Approach to Recognizing Event Occurrences in Distributed Computations," *Proc. IEEE 8th Int'l Conf. on Dist. Comput. Sys.*, 1988, pp. 300-307.

[20] M. Steenstrup, Routing in Communications Networks, Prentice Hall, Inc. 1995.

[21] D. Wang and J. Cao, "On optimal hierarchical configuration of distributed system on mesh and hypercube," *International Journal of Foundations of Computer Science*, Vol. 15, No. 3, pp. 517-534, June 2004.

[22] C.-Q. Yang and B.P. Miller, "Performance Measurement for Parallel and Distributed Programs: A Structured and Automatic Approach," *IEEE Transactions on Software Engineering*, Vol. 15, No. 12, pp. 1615-1629, December 1989.

[23] Y. Zhu, "Efficient Processor Allocation Strategies for Mesh-Connected Parallel Computers," *Journal of Parallel and Distributed Computing*, No. 16, pp. 328-337, 1992.

[24] F. Zhang and J. Cao, "Hierarchical Configuration of Monitoring Units in a Tree-structured Distributed System," *Proc. 1999 IEEE International Conference on Systems, Man and Cybernetics*, Japan, Sep., 1999.