Fault-Tolerance Schemes for Hierarchical Mesh Networks

Jason Zurawski {zurawski@eecis.udel.edu}
Dajin Wang {wang@pegasus.montclair.edu}

Department of Computer Science
Montclair State University
Upper Montclair, NJ 07043, USA

1 Introduction

In [3], a hierarchical configuration for mesh was developed. The proposed scheme divides a mesh network into uniform smaller clusters. Each of these clusters contains a *leader* (or *monitor*) to communicate with the other members of the group. Leaders are then required to communicate with other leaders to form groups at higher levels. The hierarchical approach has been shown to reduce the communication cost by reducing the overall distance traveled by messages in the network [1, 2, 3]. Experiments were conducted on the hierarchical configuration to simulate different activities that it may be used for. Simulations were designed to test various sizes of the underlying mesh, as well as potential cluster sizes that may be utilized. In efforts to see if additional improvements could be made, a variety of throughputs of data were tested for the system.

As always in any system consisting of a large number of elements, the robustness of such a hierarchically configured system is an important issue. When certain processing units are chosen as leaders, they become very important to the whole system. If a leader were to experience a failure, the localized cluster data may fail to reach the upper levels of the hierarchy. Failures at higher levels are even more disturbing to the effectiveness of the network system. The work in this paper explores possible techniques and algorithms to help a hierarchically configured mesh deal with failure of nodes. It will show that even though failures have occurred, it is still possible to achieve an overall good performance through the use of certain heuristic approaches. Individual processing unit locations will be examined to determine their overall impact to clusters as well as the system as a whole.

We propose two algorithms to deal with failure of cluster leaders. The first algorithm shifts the leader of individual clusters as needed, thus minimizing the cost within each of the clusters. The second algorithm involves shifting the leaders of the entire configuration to better suited nodes, thus minimizing the costs at higher levels. These two strategies will be evaluated/compared for speed, accuracy, and optimality through various empirical tests.

The rest of this paper is organized as follows. In Section 2 we discuss various situations of node failure in a mesh. Different strategic locations within the mesh will be presented, as well as techniques for dealing with their failure. A single-level mesh will be examined first, to illustrate basic ideas; these ideas will then be extended to a two-level hierarchy. Section 3 will present two different algorithms designed to help deal with the failure of processing units in a hierarchical mesh. Section 4 will summarize the paper, and draw conclusions based on the findings.

2 Node failure within mesh

We classify mesh nodes into two types. *Leaf nodes* exist on the border of the mesh, and can have either two or three connections to their neighbors. "Corner" and "edge" nodes are of this type. *Internal nodes* exist within the borders of the mesh, and have four connections to neighboring nodes. These two types of processing units will produce different results when they experience failure. If a failure occurs to a leaf node it will not burden the system with additional communication cost; the faulty node can be simply thought of as being removed from the system. If an internal node fails, it will cause the system to incur more communication cost by forcing other nodes to take longer paths to the monitor. In the event that several nodes of close proximity fail, a limited number of paths to the monitor could remain. A situation like this could easily result in much higher communication cost.

A special case of internal nodes is it being a monitor. If a level monitor experiences a failure it becomes necessary to elect a new one. Electing the new monitor will be dealt with in two cases: having an N size that is even or odd. These situations will be examined first in the context of a single level mesh. The ideas will then be extended to a hierarchically configured system of two levels.

2.1 Monitor node failure within a single level mesh

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. A single level of mesh uses a centrally positioned monitor to minimize the communication cost. We treat an even and odd mesh separately due to the design of the "central area." The ultimate goal is to keep the cost of a faulty mesh similar to that of the ideal mesh (i.e., mesh without faults). When moving the monitor it is possible to either gain or lose cost, depending on the position of the new monitor. The strategies described below aim to maintain an optimal "near-central" choice in order to minimize the cost.

2.1.1 N is even

When N is even, a true central node does not exist. Any one of the four nodes in the central "area" of the mesh may be picked as the monitor. So we just pick the top-left node to

0-7695-2405-2/05 $20.00 © 2005 IEEE

be the standard choice. The darkest node in Figure 1 shows an instance. When this monitor fails to function, we should choose one of the three remaining nodes from the central region to be the new monitor. There are basically two ways to choose the replacement: choosing the node diagonal to the failed node, or choosing the nodes directly below or next to the failed node. The two choices are not at all equal; the first choice will benefit the system while the second choice causes the mesh to incur additional communication cost.

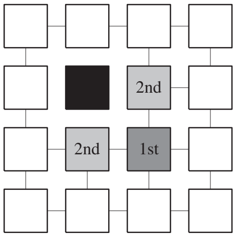


Figure 1. Two possible choices for replacement when a failure occurs. The first choice is the node diagonal to the failed node; the second choice is the node below or next to the failed one.

The diagonal node is considered a better choice for our new monitor. Choosing it will yield a net loss of 2 for the communication cost. Refer to Figure 2. The communication cost assignment has not changed from that of an ideal system, with the exception of having to remove the single failed node. The diagonal choice is preferred also because the new monitor (the grey node in Figure 2) is able to retain all its four active connections. The second choice, in contrast, offers 3 active connections.

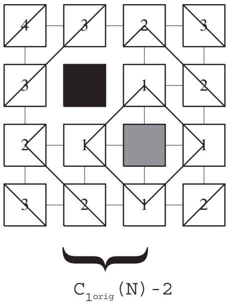


Figure 2. Recalculation of cost, for a mesh of $N = 4$, when we choose the diagonal node in the event of monitor failure.

We can utilize the original formula obtained in [3] for calculating the cost of an even mesh. Since the replaced and the replacing nodes are actually symmetrical to each other, the cost assignments are also symmetrical, except that the failed monitor now does not incur any cost, which would be 2. Therefore the total cost for choosing the diagonal node as the new monitor, denoted as $C_{e_1}(N)$, is calculated as follows.

$$C_{e_1}(N) = C_{e_{orig}}(N) - 2 = \frac{N^3}{2} - 2 = \frac{N^3 - 4}{2}$$

Figure 2 shows the new cost assignment. The black node is the failed monitor, causing the grey node to be chosen as the new monitor. The concentric diamonds indicate the nodes of the same cost. The only difference between this system's cost and that of an ideal system is the loss of the old monitor.

If the diagonal node is not available it becomes necessary to pursue other options. The nodes directly below or to the right of the failed node will produce the same result: a total gain of $N - 3$ in communication cost. The reason for the extra cost is due to the "detour" taken by all nodes above (or left to) the failed monitor. As shown in Figure 3, the concentric diamonds take on a new form. The failed node now sits between the new monitor and all other nodes vertically above it. All these nodes will experience an extra distance of 2. The failed monitor does not incur any cost, which would be 1.

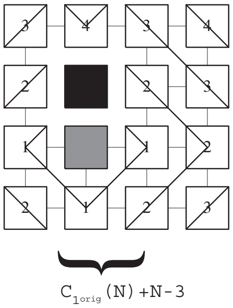


Figure 3. Cost assignment for an even mesh of $N = 4$, when we choose the node next to, or below the failed node as the new monitor. Note all nodes above the failed one (black) will travel an extra distance of 2 to reach the new monitor (grey).

The mesh will experience a total cost gain of $2 \times (\frac{N}{2} - 1)$, and a loss of 1 when we choose this replacement. The cost, denoted as $C_{e_2}(N)$, can be obtained again making use of the formula from [3]:

$$C_{e_2}(N) = C_{e_{orig}}(N) + 2\left(\frac{N}{2} - 1\right) - 1 = \frac{N^3 + 2N - 6}{2}$$

Summarizing the preceding discussion, for an even N, the total communication cost $C_e(N)$ for an $N \times N$ mesh when replacing a failed monitor is:

$$C_e(N) = \begin{cases} \frac{N^3-4}{2}, & \text{diagonal replacement} \\ \frac{N^3+2N-6}{2}, & \text{belwo/right replacement} \end{cases} \qquad (1)$$

2.1.2 N is odd

When N is odd, there exists a true central processing unit to serve as the monitoring node. If this node fails, any of the nodes that surround the failed node can be used as a replacement monitor, yielding a net gain of $N + (N - 2)$ in communication cost. We experience the gain due to detour routes around the failed node(s). Although any of the nodes can be picked to get the same result, there are two distinct categories to choose from: nodes to the diagonal of the failed node, or nodes next to the failed node, as shown in Figure 5.

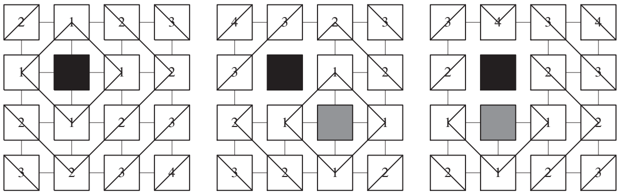


Figure 4. Summary of failure replacement strategies for even-sized mesh. On the far left is the mesh without monitor failure. Next to it are the two strategies to replace the failed monitor.

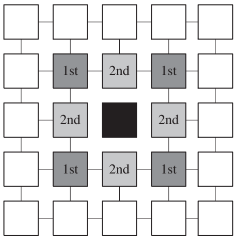


Figure 5. The "true center" of an odd mesh allows for two possible categories of choices when a failure occurs. Any of the choices will produce the same final result in terms of added cost.

To calculate the added cost, refer to Figure 6. we first split the odd mesh into an even mesh of size $N-1$, and then add an extra "gray area" to the previous total as seen in Figure 6. The concentric diamonds that surround the monitor are un-broken, but additional cost is incurred due to the communication cost of nodes in the "gray area."

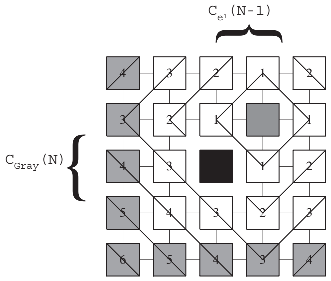


Figure 6. Cost breakdown for an odd mesh of $N = 5$, when we choose the diagonal node in the event of a failure.

Using the $C_{e_1}(N-1)$ formula that is a part of Equation 1, $C_{o_1}(N)$ can be calculated as follows.

$$C_{o_1}(N) = C_{e_1}(N-1) + C_{Gray}(N)$$

But

$$C_{Gray}(N) = (1 \cdot (N+1)) + \left(2 \cdot \frac{N+1}{2}\right) + (2 \cdot N)$$

$$+ \left(4 \sum_{i=\frac{N-1}{2}-1}^{N-1} i\right) = \frac{3N^2+1}{2}$$

Therefore

$$\begin{aligned} C_{o_1}(N) &= C_{e_1}(N-1) + C_{Gray}(N) \\ &= \left(\frac{(N-1)^3-4}{2}\right) + \frac{3N^2+1}{2} \\ &= \frac{N^3+3N-4}{2} \end{aligned}$$

When choosing the nodes that lie in a straight line from the failed node, we use a similar approach to the calculation of $C_{o_1}(N)$: we use $C_{e_2}(N-1)$ from Equation (1) to serve as the odd mesh, as well as add in another "gray area" that surrounds it, as shown in Figure 7.

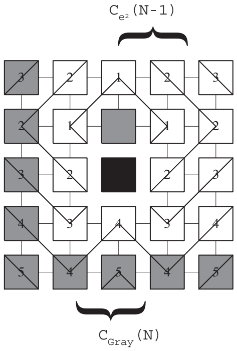


Figure 7. Cost breakdown for a odd mesh of $N = 5$, when we any of the other nodes in the event of a failure.

The calculation is as follows:

$$C_{o_2}(N) = C_{e_2}(N-1) + C_{Gray}(N)$$

$$\begin{aligned} C_{Gray}(N) &= (N-1) + \left(\frac{N-1}{2}\right) + (2 \cdot N) \\ &+ \left(\frac{N+1}{2} + 2\right) + \left(\sum_{i=\frac{N+1}{2}+1}^{N-1} 2i-1\right) \\ &= \frac{3N^2-2N+5}{2} \end{aligned}$$

$$C_{o_2}(N) = C_{e_2}(N-1) + C_{Gray}(N) = \frac{N^3+3N-4}{2}$$

Summarizing the preceding discussion, for an odd N, the total communication cost $C_o(N)$ for an $N \times N$ mesh when replacing a failed monitor is:

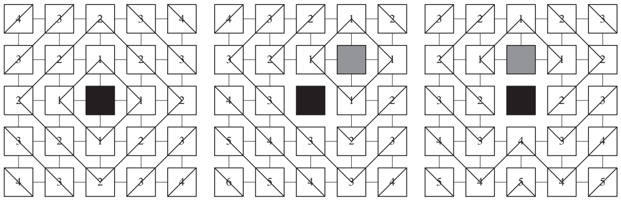


Figure 8. Summary of failure replacement strategies for odd-sized mesh. On the far left is the mesh with a functioning monitor. In the middle and on the right are the two strategies to replace the failed monitor.

$$C_o(N) = \frac{N^3 + 3N - 4}{2} \tag{2}$$

As shown in the even and odd calculations, the nodes that are diagonal to a failed node are the most sought after. This is due to the fact that they have more active connections (i.e., a maximum of 4) than that of the other nodes (i.e., a maximum of 3). Future algorithms will attempt to take advantage of this property.

2.2 Hierarchically configured faulty mesh

Hierarchical configuration of a system with faulty nodes is largely similar to that of a fault-free system, with the additional issue of incorporating the node replacement strategies discussed in Section 2.1. The idea, as stated previously, is to divide the entire mesh into identical submeshes. The submeshes will each contain a monitor, and each monitor will be a member of the upper level mesh. The upper level will have a monitor of its own. We again choose a sub mesh size of μ, a factor of N in the closest neighborhood of $\sqrt[3]{2N}$, so we can achieve the minimum communication cost.

Let the sub mesh be of dimension $\mu \times \mu$, and suppose the original monitor failed. Using the replacement strategies discussed in the preceding section, by equations (1) and (2) the cost for local monitoring will be:

$$C(\mu) = \begin{cases} \frac{\mu^3+3\mu-4}{2}, & \mu \text{ odd} \\ \frac{\mu^3-4}{2}, & \mu \text{ even}_1 \text{ (diagonal replacement)} \\ \frac{\mu^3+2\mu-6}{2}, & \mu \text{ even}_2 \text{ (belwo/right replacement)} \end{cases}$$

Therefore the total cost for all $(\frac{N}{\mu})^2$ level-one sub meshes can be calculated as:

$$C_I(N,\mu) = \begin{cases} \frac{\mu^3+3\mu-4}{2} \cdot (\frac{N}{\mu})^2, & \mu \text{ odd} \\ \frac{\mu^3-4}{2} \cdot (\frac{N}{\mu})^2, & \mu \text{ even}_1 \\ \frac{\mu^3+2\mu-6}{2} \cdot (\frac{N}{\mu})^2, & \mu \text{ even}_2 \end{cases} \tag{3}$$

At the higher level, note that all local monitors form a mesh by themselves. Choosing the central or near-central node among them as the monitor will give the minimum communication cost. However, the cost of "one step" (i.e., passage of data from a node to its immediate neighbor) is μ instead of 1. Applying equations (1) and (2) again, the cost for level-two, if replacing a failed monitor, is given as follows:

$$C_{II}(N,\mu) = \begin{cases} \frac{(\frac{N}{\mu})^3+3\frac{N}{\mu}-4}{2} \cdot \mu, & \frac{N}{\mu} \text{ odd} \\ \frac{(\frac{N}{\mu})^3-4}{2} \cdot \mu, & \frac{N}{\mu} \text{ even}_1 \\ \frac{(\frac{N}{\mu})^3+2\frac{N}{\mu}-6}{2} \cdot \mu, & \frac{N}{\mu} \text{ even}_2 \end{cases} \tag{4}$$

Combining equations (3) and (4), we have the expression for total cost of the two-level hierarchical monitoring system if we do replacement at all levels of hierarchy:

$$C_{total}(N,\mu) = C_I(N,\mu) + C_{II}(N,\mu)$$

$$= \begin{cases} \frac{\mu^3(N^2-4)+3N\mu^2+3N^2\mu+N^2(N-4)}{2\mu^2}, & \mu \text{ odd}, \frac{N}{\mu} \text{ odd} \\ \frac{\mu^3(N^2-4)+3N^2\mu+N^2(N-4)}{2\mu^2}, & \mu \text{ odd}, \frac{N}{\mu} \text{ even}_1 \\ \frac{\mu^3(N^2-4)+2N\mu^2+3N^2\mu+N^2(N-4)}{2\mu^2}, & \mu \text{ odd}, \frac{N}{\mu} \text{ odd}_2 \\ \frac{\mu^3(N^2-4)+3N\mu^2N^2(N-4)}{2\mu^2}, & \mu \text{ even}_1, \frac{N}{\mu} \text{ odd} \\ \frac{\mu^3(N^2-4)+N^2(N-4)}{2\mu^2}, & \mu \text{ even}_1, \frac{N}{\mu} \text{ even}_1 \\ \frac{\mu^3(N^2-6)+2N\mu^2+N^2(N-4)}{2\mu^2}, & \mu \text{ even}_1, \frac{N}{\mu} \text{ even}_2 \\ \frac{\mu^3(N^2-4)+3N\mu^2+2N^2\mu+N^2(N-6)}{2\mu^2}, & \mu \text{ even}_2, \frac{N}{\mu} \text{ odd} \\ \frac{\mu^3(N^2-4)+2N^2\mu+N^2(N-6)}{2\mu^2}, & \mu \text{ even}_2, \frac{N}{\mu} \text{ even}_1 \\ \frac{\mu^3(N^2-6)+2N\mu^2+2N^2\mu+N^2(N-6)}{2\mu^2}, & \mu \text{ even}_2, \frac{N}{\mu} \text{ even}_2 \end{cases}$$

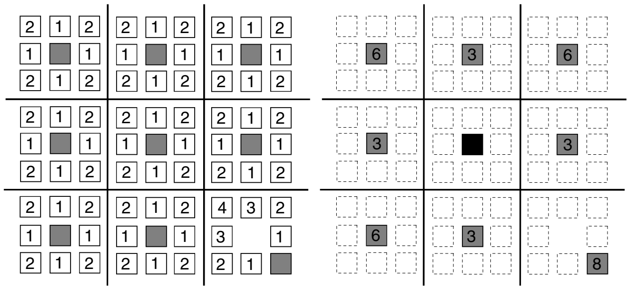
3 Failure recovery schemes

When dealing with node failure in the hierarchically structured mesh system, we need to worry about failure within an individual clusters as well as failure in the higher level. A single failed node can have a impact upon both logical levels in the hierarchy. When our primary concern is failure within a cluster, we resorted to *singular shifting* strategy. If keeping regularity at the higher levels is more highly desired, *total shifting* will be the preferred strategy. These two methods will be discussed in details below, in an effort to determine which method is better for a given situation.

3.1 Singular shifting

The greedy principal is the main rationale for the conception of *singular shifting*. Our communication model is that within a cluster, all nodes communicate with the monitor node. In this paper, we do not consider the failure of non-monitor nodes since they do not affect the hierarchical structure in a significant way.

Singular shifting algorithm only replaces the monitor in the cluster where failure occurs. If a monitor fails, a new monitor will be chosen using the strategies proposed in last section. This process is carried out for each cluster that has failed monitor, as well as at the upper level. A simple example is shown in Figure 9. In Figure 9(a), the monitor of the lower-right cluster failed. By our replacement strategy, the node diagonal to it becomes the new monitor. The new upper level structure is shown in Figure 9(b). The grey node and all nodes with costs form the upper level "pseudo mesh." Note that the total communication cost is changed and needs to be recalculated.



(a) Lower level of hierarchy (b) Upper level of hierarchy

Figure 9. A faulty 9×9 mesh with cluster size of $\mu = 3$. The failed monitor in the lower-right cluster is taken away. The monitor nodes (including the replacing monitor) are in gray. Communication costs are listed for each cluster at the lower level (a), as well as for the monitors at the upper level (b).

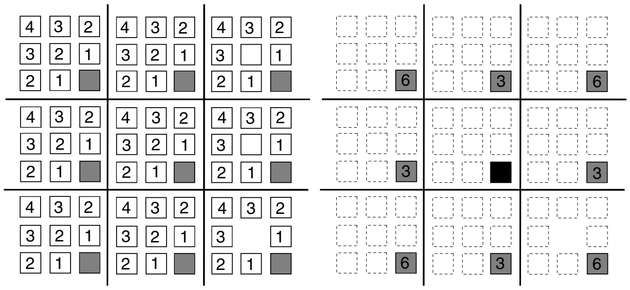
As was pointed out earlier, this algorithm aims at keeping optimality in each cluster. The upper level therfore may experience sub-optimal performance due to the shifting of monitors. *Singular shifting* does not make an attempt to preserve the system's regularity at the upper level.

3.2 Total shifting

When the hierarchical mesh experiences more than a few monitor failures, all failed cluster heads need to be replaced. This means that cluster heads may be in different relative positions, resulting in longer routes at the upper level. On the other hand, having a uniformity at the upper level is always desirable, especially when large amounts of data must be transfered. So it would be beneficial to keep the upper level uniform, as well as keeping distances to a minimum. This principal motivates the second algorithm for dealing with fault tolerance: *total shifting*.

In *total shifting*, if any of the monitors has failed, we perform the same operation on all cluster heads. A final replacement position has to be applicable to all clusters. When such a position is determined we move *every* cluster's monitor to that position, regardless of whether the original monitor is faulty or fault-free.

With this scheme we aim at preserving the regular structure at the upper level. An example is shown in Figure 10. There occurs a failed monitor in the lower-right cluster. All other clusters are fault-free. It is then determined that the node diagonal to the failed one be the new monitor. The replacement is carried out in all clusters, as shown in Figure 10(a). Figure 10(b) shows the resultant upper level.



(a) Lower level of hierarchy (b) Upper level of hierarchy

Figure 10. A faulty 9×9 mesh with cluster size of $\mu = 3$. The failed monitor in the lower-right cluster is removed. All clusters replace their monitors as in the failed one, to maintain the regularity at the upper level. Communication costs are listed for each cluster at the lower level (a); as well as at the upper level (b).

Both *singular shifting* and *total shifting* attempt to claim back as much as possible the optimality lost when failure within the mesh has occured. Evaluation of both schemes for various situations have been performed to reveal where the strong and weak points lie for each of them.

4 Conclusion

We have proposed fault-tolerating schemes for hierarchical meshes. The performance of the proposed schemes were evaluated through simulation. Simulation results show that for moderate levels of failure (i.e., less than or equal to 20%), the proposed schemes can achieve performance similar to that of an fault-free system.

Many more interesting problems can be raised from this initial research. For one example, what is a proper circumstance to apply *singular shifting* vs. *total shifting*? It is conjectured that as the system carries large amounts of data from the clusters to the higher level, *total shifting* would be more desirable. If the amount of data is smaller at the higher levels, and larger within the clusters, *singular shifting* will produce better results. A research further defining the applicability of proposed schemes in a quantified way will be our future work.

References

[1] J. Cao, K. Zhang, and O. de Vel, "On Heuristics for Optimal Configuration of Hierarchical Distributed Monitoring Systems," *Journal of Systems and Software*, Vol. 45, No. 2, pp. 141-154, 1999.

[2] 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.

[3] 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.